CSCI567 Machine Learning (Spring 2025)
Instructor: Haipeng Luo

Sample Quiz Two

Duration: 140 minutes

NAME _____

Stu ID _____

Instructions:

1. Please fill in your name and ID in the space above.
2. When the exam ends, you have 20 minutes to upload your solutions to Gradescope using your phone (**do not use your phone for any reason before that!**). Please also turn in your exam before leaving the room. **Do not continue working on the exam during these last 20 minutes.**
3. If you encounter any difficulty in uploading (or you are taking the quiz in a test center that prohibits cell phones), simply turn in your exam and we will upload it for you.
4. This exam has a total of 16 pages (including this cover page). Check and make sure that you are not missing any pages. Additionally, there are 5 pages provided for scratch work at the end.
5. Any kind of cheating will be reported and lead to a 0 score for the entire exam.

1 Multiple-Choice Questions (30 points)

IMPORTANT: *Select ALL answers that you think are correct. You get 0.5 point for selecting each correct answer and similarly 0.5 point for not selecting each incorrect answer.*

(1) Which of the following is a possible application of clustering?

(A) Compressing images.
(B) Discovering communities in a social network.
(C) Accelerating downstream algorithms.
(D) Finding similar customers in market research.

Answer: ABCD

(2) Which of the following about the convergence of K-means is correct?

(A) K-means always finds the global minimum of the K-means objective, but it might take very long time.
(B) K-means always finds a local minimum of the K-means objective.
(C) K-means++ always finds the global minimum of the K-means objective, but it might take very long time.
(D) In expectation K-means++ finds a good approximation of the global minimum of the K-means objective.

Answer: BD

(3) Which of the following about Gaussian Mixture Model (GMM) is correct?

(A) GMM assumes that the data are generated stochastically in a particular way, and thus can only be applied if we know the training data are indeed generated from this model.
(B) The MLE of a GMM model can be found using the EM algorithm.
(C) Learning a GMM via EM gives not only the cluster (soft) assignments and centers, but also other parameters such as mixture weights and covariance matrices.
(D) GMM for clustering always performs better than K-means since it learns more parameters.

Answer: C

(4) Which of the following about density estimation is correct?

(A) Density estimation is an unsupervised learning problem.
(B) Density estimation always requires assuming a generative model to start with.
(C) Kernel density estimation has no hyperparameters.
(D) Kernel density estimation is a non-parametric method.

Answer: AD

(5) Which of the following about Naive Bayes is correct?

(A) Naive Bayes assumes that different features are all independent of each other.
(B) Naive Bayes with a Gaussian model learns a linear classifier if the variance is fixed.
(C) Naive Bayes with a Gaussian model learns a quadratic classifier if the variance is also a parameter.

(D) Naive Bayes and logistic regression learn the same model via different methods.

Answer: BC

(6) Which of the following about Principal Component Analysis (PCA) is correct?

(A) PCA can be used to compress a dataset.
(B) PCA is useful for visualizing a dataset.
(C) PCA requires finding all eigenvectors of the covariance matrix.
(D) PCA requires finding all eigenvalues if we want to find enough principal components to cover a certain percentage of the spectrum.

Answer: AB

(7) Let $\boldsymbol{X}$ be an $N \times D$ data matrix where each row corresponds to a portrait, and $\boldsymbol{V} \in \mathbb{R}^{D \times p}$ be the top p eigenfaces. Which of the following is the (lossy) reconstruction of the original portraits?

(A) $\boldsymbol{V}$ (B) $\boldsymbol{X}\boldsymbol{V}$ (C) $\boldsymbol{X}\boldsymbol{V}\boldsymbol{V}^\top$ (D) $\boldsymbol{X}^\top\boldsymbol{X}$

Answer: C

(8) Suppose that the covariance matrix of a dataset $\boldsymbol{X} \in \mathbb{R}^{N \times 4}$ is $\boldsymbol{X}^\top\boldsymbol{X} = \begin{pmatrix} 25 & 3 & 13 & -1 \\ 3 & 25 & -1 & 13 \\ 13 & -1 & 25 & 3 \\ -1 & 13 & 3 & 25 \end{pmatrix}$ and its top three eigenvalues are $40, 36$ and 16 respectively. How many principal components we need to pick when performing PCA if we want 80% of the variance explained?

(A) 1 (B) 2 (C) 3 (D) 4

Answer: C. This is because $\frac{40+36}{25+25+25+25} \leq 80\%$ and $\frac{40+36+16}{25+25+25+25} \geq 80\%$. No eigendecomposition needed.

(9) Which of the following about kernel PCA is correct?

(A) Kernel PCA requires centering the original dataset.
(B) Kernel PCA requires rescaling the L2 norm of an eigenvector to $1/\sqrt{\lambda}$ where λ is the corresponding eigenvalue.
(C) Running kernel PCA is always slower than running standard PCA.
(D) The output of kernel PCA is not a linear transformation of the original dataset.

Answer: BD

(10) Suppose that $z_1^*, \ldots, z_T^*$ is the output of the Viterbi algorithm given a sequence of observations $x_1, \ldots, x_T$. Which of the following is NOT correct?

(A) z_1^* is the most likely first state given $x_1, \ldots, x_T$.
(B) z_1^* is the most likely first state given x_1.
(C) z_T^* is the most likely last state given $x_1, \ldots, x_T$.
(D) z_T^* is the most likely last state given x_T.

Answer: ABCD

(11) Which of the following about Recurrent Neural Networks (RNN) and Transformers is correct?

(A) The time complexity for RNN to process a sequence of length T is $O(T^2)$.
(B) Doubling the number of layers of an RNN also roughly doubles its number of parameters.
(C) Layer normalization and batch normalization are the same.
(D) When generating texts via softmax, the larger the temperature parameter is, the more random the final outputs are.

Answer: BD

(12) In a multi-armed bandit problem with two arms and binary rewards, suppose that we have selected the first arm 6 times, 3 of which yield reward 1, and the second arms 3 times, 2 of which yield reward 1. Which of the following about the behavior of the UCB algorithm in the next round is correct?

(A) UCB selects the second arm because $\frac{1}{2} + 2\sqrt{\frac{\ln 10}{6}} < \frac{2}{3} + 2\sqrt{\frac{\ln 10}{3}}$.
(B) UCB selects the second arm because $\frac{1}{2} + 2\sqrt{\frac{\ln 10}{3}} < \frac{2}{3} + 2\sqrt{\frac{\ln 10}{2}}$.
(C) UCB selects the second arm because $\frac{1}{2} + 2\sqrt{\frac{\ln 5}{3}} < \frac{2}{3} + 2\sqrt{\frac{\ln 5}{2}}$.
(D) UCB selects the second arm with probability $\frac{2}{3}$.

Answer: A

(13) Which of the following about reinforcement learning is correct?

(A) Reinforcement learning with S states is simply a combination of S multi-armed bandit problems.
(B) Value iteration always converges, but there is no guarantee on how long it will take.
(C) Both model-based and model-free methods need to address the exploration-exploitation trade-off.
(D) Q-learning learns the Q function directly without estimating the model parameters.

Answer: CD

(14) Which of the following is the correct Value Iteration update?

(A) $V(s) \leftarrow \min_{a\in\mathcal{A}} \left(r(s,a) + \gamma \sum_{s'\in\mathcal{S}} P(s'|s,a)V(s')\right)$
(B) $V(s) \leftarrow \max_{a\in\mathcal{A}} \left(r(s,a) + \gamma \sum_{s'\in\mathcal{S}} P(s'|s,a)V(s')\right)$
(C) $V(s) \leftarrow \max_{a\in\mathcal{A}} \left(r(s,a) + \gamma \sum_{s'\in\mathcal{S}} P(s|s',a)V(s')\right)$
(D) $V(s) \leftarrow \max_{a\in\mathcal{A}} r(s,a) + \gamma \max_{a\in\mathcal{A}} \left(\sum_{s'\in\mathcal{S}} P(s'|s,a)V(s')\right)$

Answer: B

(15) Policy gradient methods directly find the optimal policy by applying (stochastic) gradient descent to the objective function $R(\pi_\rho)$. Which of the following is equal to $\nabla_\rho R(\pi_\rho)$?

(A) $\mathbb{E}_\tau \left[\sum_{h=1}^{H} \nabla_\rho \log \pi_\rho(a_h|s_h) R(\tau)\right]$.
(B) $\mathbb{E}_\tau \left[\sum_{h=1}^{H} \nabla_\rho \log \pi_\rho(a_h|s_h) \left(R(\tau) - 1\right)\right]$.
(C) $\mathbb{E}_\tau \left[\sum_{h=1}^{H} \nabla_\rho \log \pi_\rho(a_h|s_h) \left(R(\tau) - r(s_h,a_h)\right)\right]$.
(D) $\mathbb{E}_\tau \left[\sum_{h=1}^{H} \nabla_\rho \log \pi_\rho(a_h|s_h) \left(\sum_{h'=h}^{H} r(s_{h'},a_{h'}) - V_\theta(s_h)\right)\right]$ for some target network θ.

Answer: ABD

2 Clustering and Kernel (12 points)

Consider applying K-means++ to cluster a dataset of N samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^D$. Instead of doing this directly, we apply a feature map $\boldsymbol{\phi} : \mathbb{R}^D \to \mathbb{R}^M$ to each example, and then apply K-means++. With a corresponding kernel function $k(\cdot, \cdot)$ for this feature map, you need to follow the steps below to show that this algorithm can be efficiently implemented (that is, without operating in space $\mathbb{R}^M$).

2.1 First, recall that in the initialization step of K-means++, we need to randomly select K centers based on squared L2 distances, which requires calculating $\|\boldsymbol{\phi}(\boldsymbol{x}_n) - \boldsymbol{\phi}(\boldsymbol{x}_m)\|_2^2$ for some $n, m \in [N]$. Express this quantity using the kernel function only (show your derivation). (2 points)

$$\begin{aligned} \|\boldsymbol{\phi}(\boldsymbol{x}_n) - \boldsymbol{\phi}(\boldsymbol{x}_m)\|_2^2 &= \boldsymbol{\phi}(\boldsymbol{x}_n)^\top \boldsymbol{\phi}(\boldsymbol{x}_n) - 2\boldsymbol{\phi}(\boldsymbol{x}_n)^\top \boldsymbol{\phi}(\boldsymbol{x}_m) + \boldsymbol{\phi}(\boldsymbol{x}_m)^\top \boldsymbol{\phi}(\boldsymbol{x}_m) && \text{(1 point)} \\ &= k(\boldsymbol{x}_n, \boldsymbol{x}_n) - 2k(\boldsymbol{x}_n, \boldsymbol{x}_m) + k(\boldsymbol{x}_m, \boldsymbol{x}_m) && \text{(1 point)} \end{aligned}$$

2.2 Suppose that $\mathcal{S} \subset [N]$ contains a nonempty subset of examples belonging to the same cluster in some iteration of K-means++. In the next iteration, one needs to compute the squared distance between an arbitrary example $\boldsymbol{\phi}(\boldsymbol{x}_n)$ and the center of this cluster $\boldsymbol{\mu} = \frac{1}{|\mathcal{S}|} \sum_{m \in \mathcal{S}} \boldsymbol{\phi}(\boldsymbol{x}_m)$, that is, $\|\boldsymbol{\phi}(\boldsymbol{x}_n) - \boldsymbol{\mu}\|_2^2$. Once again, express this quantity using the kernel function only (show your derivation). (4 points)

$$\begin{aligned} \|\boldsymbol{\phi}(\boldsymbol{x}_n) - \boldsymbol{\mu}\|_2^2 &= \boldsymbol{\phi}(\boldsymbol{x}_n)^\top \boldsymbol{\phi}(\boldsymbol{x}_n) - 2\boldsymbol{\phi}(\boldsymbol{x}_n)^\top \boldsymbol{\mu} + \boldsymbol{\mu}^\top \boldsymbol{\mu} && \text{(1 point)} \\ &= k(\boldsymbol{x}_n, \boldsymbol{x}_n) - \frac{2}{|\mathcal{S}|} \sum_{m \in \mathcal{S}} \boldsymbol{\phi}(\boldsymbol{x}_n)^\top \boldsymbol{\phi}(\boldsymbol{x}_m) + \frac{1}{|\mathcal{S}|^2} \sum_{m, m' \in \mathcal{S}} \boldsymbol{\phi}(\boldsymbol{x}_m)^\top \boldsymbol{\phi}(\boldsymbol{x}_{m'}) && \text{(2 points)} \\ &= k(\boldsymbol{x}_n, \boldsymbol{x}_n) - \frac{2}{|\mathcal{S}|} \sum_{m \in \mathcal{S}} k(\boldsymbol{x}_n, \boldsymbol{x}_m) + \frac{1}{|\mathcal{S}|^2} \sum_{m, m' \in \mathcal{S}} k(\boldsymbol{x}_m, \boldsymbol{x}_{m'}) && \text{(1 point)} \end{aligned}$$

2.3 Based on your answers from the last two questions, fill in the missing details in Algorithm 1. More specifically,

- complete the for loop in Line 7 which finds the initial center indices $n_2, \dots, n_K \in [N]$;
- complete the for loop in Line 15 which finds the new partition $\mathcal{S}'_1, \dots, \mathcal{S}'_K$ based on $\mathcal{S}_1, \dots, \mathcal{S}_K$.

Note that we have pre-computed the Gram matrix $\boldsymbol{M}$ as an input, so you can directly use $M_{n,m}$ whenever you need $k(\boldsymbol{x}_n, \boldsymbol{x}_m)$. (6 points)

Algorithm 1: K-means++ with kernel

1 **Input:** dataset $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}$ with Gram matrix $\boldsymbol{M}$
2 **Output:** a partition of the dataset $\mathcal{S}_1, \dots, \mathcal{S}_K \subset [N]$
3 **Initialize:**
5 Uniformly at random select an index $n_1 \in [N]$ as the first center.
7 **for** $k = 2, \dots, K$ **do**
8 Randomly select the k-th center's index n_k such that

$$\mathbb{P}[n_k = n] \propto \min_{j=1,\dots,k-1} \left(M_{n,n} - 2M_{n,n_j} + M_{n_j,n_j} \right)$$

10 Set $\mathcal{S}_k = \{n_k\}$ for all $k \in [K]$
11 **Repeat:**
13 Set $\mathcal{S}'_k = \emptyset$ for all $k \in [K]$ ▷ initialize the partition as empty sets
15 **for** $n = 1, \dots, N$ **do**
16 $k = \operatorname{argmin}_{j \in [K]} \left(M_{n,n} - \frac{2}{|\mathcal{S}_j|} \sum_{m \in \mathcal{S}_j} M_{n,m} + \frac{1}{|\mathcal{S}_j|^2} \sum_{m,m' \in \mathcal{S}_j} M_{m,m'} \right)$
17 $\mathcal{S}'_k \leftarrow \mathcal{S}'_k \cup \{n\}$
19 Set $\mathcal{S}_k = \mathcal{S}'_k$ for all $k \in [K]$ ▷ overwrite $\mathcal{S}_1, \dots, \mathcal{S}_K$ with the new partition.

Rubrics: 2 points for the first loop and 4 points for the second loop. Note that dropping $M_{n,n}$ in both places results in the same algorithm. Do not deduct points for mistakes inherited from the last two questions.

3 Naive Bayes Classifiers (12 points)

Suppose that we have the following training data: each sample has three features (Weather, Emotion, Homework), where Weather $\in \{Sunny, Cloudy\}$, Emotion $\in \{Happy, Normal, Unhappy\}$, and Homework $\in \{Much, Little\}$; the binary label PlayBasketball indicates whether it is suitable to play basketball. You need to build a naive Bayes classifier using this dataset. Recall that the naive Bayes assumption implies the following:

$$P(\text{Weather, Emotion, Homework} \mid \text{PlayBasketball}) =$$
$$P(\text{Weather} \mid \text{PlayBasketball}) \times P(\text{Emotion} \mid \text{PlayBasketball}) \times P(\text{Homework} \mid \text{PlayBasketball}).$$

Weather	Emotion	Homework	PlayBasketball
Sunny	Happy	Little	Yes
Sunny	Normal	Little	Yes
Cloudy	Happy	Much	Yes
Cloudy	Unhappy	Little	Yes
Sunny	Unhappy	Little	No
Cloudy	Normal	Much	No

3.1 Write down the MLE for the following parameters (no reasoning needed): (4 points)

- $P(\text{PlayBasketball} = No) = \frac{1}{3}$
- $P(\text{Weather} = Cloudy \mid \text{PlayBasketball} = Yes) = \frac{1}{2}$
- $P(\text{Emotion} = Unhappy \mid \text{PlayBasketball} = No) = \frac{1}{2}$
- $P(\text{Homework} = Little \mid \text{PlayBasketball} = Yes) = \frac{3}{4}$

Rubrics: one point each.

3.2 Given a new sample $\boldsymbol{x} = (Cloudy, Unhappy, Little)$, find out the exact value of

$$P(\text{PlayBasketball} = Yes \mid \boldsymbol{x})$$

(show your derivation), and conclude what the final prediction of the naive Bayes classifier is for $\boldsymbol{x}$. (8 points)

$$\begin{aligned}
&P(\text{PlayBasketball} = Yes \mid \boldsymbol{x}) \\
&\propto P(\text{PlayBasketball} = Yes, \boldsymbol{x}) \\
&= P(\text{PlayBasketball} = Yes) \times P(\text{Weather} = Cloudy \mid \text{PlayBasketball} = Yes) \times \\
&\quad P(\text{Emotion} = Unhappy \mid \text{PlayBasketball} = Yes) \times P(\text{Homework} = Little \mid \text{PlayBasketball} = Yes) \\
&= \tfrac{2}{3} \times \tfrac{1}{2} \times \tfrac{1}{4} \times \tfrac{3}{4} = \tfrac{1}{16}.
\end{aligned}$$

Similarly,

$$\begin{aligned}
&P(\text{PlayBasketball} = No \mid \boldsymbol{x}) \\
&\propto P(\text{PlayBasketball} = No, \boldsymbol{x}) \\
&= P(\text{PlayBasketball} = No) \times P(\text{Weather} = Cloudy \mid \text{PlayBasketball} = No) \times \\
&\quad P(\text{Emotion} = Unhappy \mid \text{PlayBasketball} = No) \times P(\text{Homework} = Little \mid \text{PlayBasketball} = No) \\
&= \tfrac{1}{3} \times \tfrac{1}{2} \times \tfrac{1}{2} \times \tfrac{1}{2} = \tfrac{1}{24}.
\end{aligned}$$

Therefore

$$P(\text{PlayBasketball} = Yes \mid \boldsymbol{x}) = \frac{\frac{1}{16}}{\frac{1}{24} + \frac{1}{16}} = \frac{3}{5},$$

which is larger than $P(\text{PlayBasketball} = No \mid \boldsymbol{x}) = 1 - \frac{3}{5} = \frac{2}{5}$, meaning that the final prediction is "Yes".

Rubrics:

- 3 points for figuring out $P(\text{PlayBasketball} = Yes \mid \boldsymbol{x}) \propto \frac{1}{16}$
- another 3 points for figuring out $P(\text{PlayBasketball} = No \mid \boldsymbol{x}) \propto \frac{1}{24}$
- 1 point for figuring out the final value of $P(\text{PlayBasketball} = Yes \mid \boldsymbol{x})$, and another 1 point for making the correct final prediction.

4 HMM and EM (22 points)

Recall that a hidden Markov model with a state space S and an observation space O is parameterized by:

- initial state distribution $P(Z_1 = s) = \pi_s$,
- transition distribution $P(Z_{t+1} = s' \mid Z_t = s) = a_{s,s'}$,
- emission distribution $P(X_t = o \mid Z_t = s) = b_{s,o}$.

Imagine a speech dataset of length-T sequences generated by this HMM. Unfortunately, due to low quality of the device that collects the speech, you only have partial observations in every sequence. Can you still learn the parameters of this HMM? Follow the steps below to do so.

4.1 Let $\Omega \subset [T]$ be a subset of time steps, and suppose that we are given a sequence of partial observations x_t for $t \in \Omega$. Further let $\Omega_{\leq t}$ be the shorthand for $\{\tau \in \Omega : \tau \leq t\}$ and similarly $\Omega_{\geq t}$ be the shorthand for $\{\tau \in \Omega : \tau \geq t\}$. Redefine the forward and backward messages as

$$\alpha_s(t) = P(Z_t = s, X_k = x_k, \forall k \in \Omega_{\leq t}) \quad \text{and} \quad \beta_s(t) = P(X_k = x_k, \forall k \in \Omega_{\geq t+1} \mid Z_t = s).$$

When the parameters of the HMM are known, computing these messages is similar to the original forward and backward procedure. Take the forward message as an example. Recall the original forward procedure:

Algorithm 2: Original forward procedure

Input: observations $x_1, \ldots, x_T$
Initialization: for each state $s \in S$, compute $\alpha_s(1) = \pi_s b_{s,x_1}$
for $t = 2, \ldots, T$ **do**
    for each state $s \in S$, compute

$$\alpha_s(t) = b_{s,x_t} \sum_{s' \in S} a_{s',s} \alpha_{s'}(t-1)$$

Fill in the missing details in the modified forward procedure below. No reasoning is needed. (4 points)

Algorithm 3: Modified forward procedure

Input: observations x_t for $t \in \Omega$
Initialization: for each state $s \in S$, compute $\alpha_s(1) = \begin{cases} \pi_s b_{s,x_1} & \text{if } 1 \in \Omega \\ \pi_s & \text{else} \end{cases}$
for $t = 2, \ldots, T$ **do**
    for each state $s \in S$, compute

$$\alpha_s(t) = \begin{cases} b_{s,x_t} \sum_{s' \in S} a_{s',s} \alpha_{s'}(t-1) & \text{if } t \in \Omega \\ \sum_{s' \in S} a_{s',s} \alpha_{s'}(t-1) & \text{else} \end{cases}$$

Rubrics: One point for each of the 4 cases.

4.2 Given the modified forward and backward messages, the next step is to compute

$$\gamma_s(t) = P(Z_t = s \mid X_k = x_k, \forall k \in \Omega) \quad \text{and} \quad \xi_{s,s'}(t) = P(Z_t = s, Z_{t+1} = s' \mid X_k = x_k, \forall k \in \Omega),$$

which is again similar to the original version. In particular, take $\gamma_s(t)$ as an example and express it using the modified $\alpha_s(t)$ and $\beta_s(t)$. Show your derivation, which can use the propositional sign, but express your final answer WITHOUT using it. (4 points)

$$\begin{aligned}
\gamma_s(t) &\propto P(Z_t = s, X_k = x_k, \forall k \in \Omega) && \text{(1 point)} \\
&= P(Z_t = s, X_k = x_k, \forall k \in \Omega_{\leq t}) P(X_{k'} = x_{k'}, \forall k' \in \Omega_{\geq t+1} \mid Z_t = s, X_k = x_k, \forall k \in \Omega_{\leq t}) && \text{(1 point)} \\
&= P(Z_t = s, X_k = x_k, \forall k \in \Omega_{\leq t}) P(X_{k'} = x_{k'}, \forall k' \in \Omega_{\geq t+1} \mid Z_t = s) \\
&= \alpha_s(t)\beta_s(t) && \text{(1 point)}
\end{aligned}$$

Therefore, the final answer is $\gamma_s(t) = \frac{\alpha_s(t)\beta_s(t)}{\sum_{s' \in S} \alpha_{s'}(t)\beta_{s'}(t)}$. (1 point)

4.3 We are now ready to apply the EM algorithm to learn this HMM. Suppose that the dataset consists of N sequences, where the n-th sequence is $x_t^{(n)}$ for $t \in \Omega^{(n)}$. In the E-step, given the statistics $\gamma_s^{(n)}(t)$ and $\xi_{s,s'}^{(n)}(t)$ computed from the last question for each sequence $n = 1, \ldots, N$ using a fixed set of HMM parameters, write down the complete log likelihood function $Q(\{\pi_s\}, \{a_{s,s'}\}, \{b_{s,o}\})$. (8 points)

For a sequence x_t for $t \in \Omega$, let q be the corresponding posterior distribution of states. Then the complete log likelihood for this sequence is

$$\begin{aligned}
&\mathbb{E}_{z_{1:T}\sim q}\left[\ln P(Z_{1:T} = z_{1:T}, X_k = x_k, \forall k \in \Omega)\right] \\
&= \mathbb{E}_{z_{1:T}\sim q}\left[\ln \pi_{z_1} + \sum_{t=1}^{T-1} \ln a_{z_t, z_{t+1}} + \sum_{t\in\Omega} \ln b_{z_t, x_t}\right] \\
&= \sum_{s\in S} \gamma_s(1) \ln \pi_s + \sum_{t=1}^{T-1} \sum_{s,s'\in S} \xi_{s,s'}(t) \ln a_{s,s'} + \sum_{o\in O} \sum_{t\in\Omega: x_t = o} \sum_{s\in S} \gamma_s(t) \ln b_{s,o}. \qquad (1)
\end{aligned}$$

Therefore, the complete log likelihood for the entire dataset is

$$Q(\{\pi_s\}, \{a_{s,s'}\}, \{b_{s,o}\}) = \sum_{n=1}^{N} \left(\sum_{s\in S} \gamma_s^{(n)}(1) \ln \pi_s + \sum_{t=1}^{T-1} \sum_{s,s'\in S} \xi_{s,s'}^{(n)}(t) \ln a_{s,s'} + \sum_{o\in O} \sum_{t\in\Omega^{(n)}: x_t^{(n)} = o} \sum_{s\in S} \gamma_s^{(n)}(t) \ln b_{s,o} \right)$$

Rubrics:

- two points for each of the three terms in Eq. (1);
- the last term in Eq. (1) can also be written as $\sum_{t\in\Omega} \sum_{s\in S} \gamma_s(t) \ln b_{s,x_t}$;
- another two points for the correct final answer that sums over all N sequences.

4.4 Complete the M-step by writing down the parameters that maximize the complete log likelihood function $Q(\{\pi_s\}, \{a_{s,s'}\}, \{b_{s,x}\})$ from the last question. No reasoning needed. (6 points)

The maximizer is

$$\pi_s \propto \sum_{n=1}^{N} \gamma_s^{(n)}(1), \quad a_{s,s'} \propto \sum_{n=1}^{N} \sum_{t=1}^{T-1} \xi_{s,s'}^{(n)}(t), \quad b_{s,o} \propto \sum_{n=1}^{N} \sum_{t \in \Omega^{(n)}: x_t^{(n)} = o} \gamma_s^{(n)}(t).$$

Rubrics: Two points for each of the three answers.

5 RNNs and Transformers (14 points)

5.1 Consider the following mini RNN (a picture taken from Lecture 11).

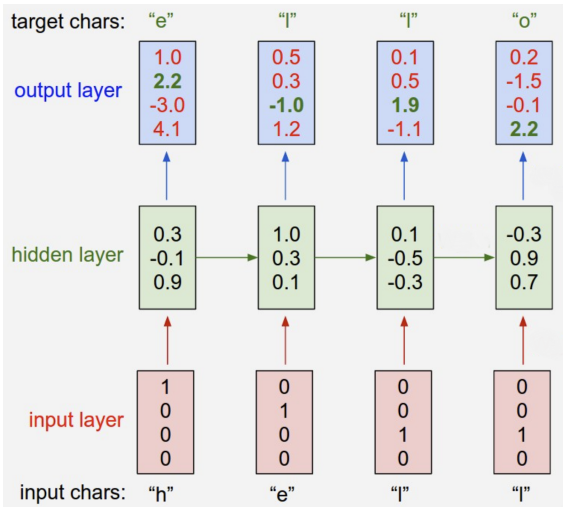

Figure 1: A mini RNN

(1) How many parameters are there in this RNN? Show your calculation and feel free to ignore all bias terms. (4 points)

Since the input/output dimension is 4 and the hidden state dimension is 3, the two matrices $\boldsymbol{W}$ and $\boldsymbol{U}$ used for the state update $\boldsymbol{h}' = \sigma(\boldsymbol{W}\boldsymbol{h} + \boldsymbol{U}\boldsymbol{x})$ are 3×3 and 3×4 respectively, while the matrix $\boldsymbol{V}$ used for the output $\boldsymbol{V}\boldsymbol{h}'$ is 4×3. Therefore, in total there are $9 + 12 + 12 = 33$ parameters.

Rubrics: 1 point for each of the three matrices, and another 1 point for the correct final answer.

(2) What is the total cross entropy loss of the first two outputs in Figure 1? Write down you answer directly using exponentials; no need to further simplify or approximate them. (4 points)

The total cross entropy loss is

$$-\ln\left(\frac{e^{2.2}}{e^{1.0}+e^{2.2}+e^{-3.0}+e^{4.1}}\right) - \ln\left(\frac{e^{-1.0}}{e^{0.5}+e^{0.3}+e^{-1.0}+e^{1.2}}\right)$$

Rubrics: Two points each.

5.2 Consider a self-attention head in an encoder of a transformer.

(1) For a 3-token input, the table below shows some partial values of the 3×3 matrix obtained after applying softmax to the attention score matrix, that is, softmax $\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d_k}}\right)$. Fill in the missing values in this table (no reasoning needed). (3 points)

0.4	0.4	0.2
0.1	0.3	0.6
0.3	0.6	0.1

Rubrics: One point each. The key is that each row (not column) forms a distribution.

(2) Continuing from the last question, suppose that the value vectors for these 3 input tokens are:

$$\begin{pmatrix} 1 \\ 2 \end{pmatrix}, \quad \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

respectively. What is the corresponding output for the 2nd token? Show your calculation. (3 points)

The output for the 2nd token should be the weighted sum of the value vectors according to the 2nd row of the matrix from the last question, that is:

$$0.1 \times \begin{pmatrix} 1 \\ 2 \end{pmatrix} + 0.3 \times \begin{pmatrix} -1 \\ 0 \end{pmatrix} + 0.6 \times \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 0.4 \\ -0.4 \end{pmatrix}$$

Rubrics: 1 point for writing down some weighted sum of the value vectors, 1 point for using the correct weights, and 1 point for the correct final answer.

6 Multi-Armed Bandits (10 points)

After using the UCB algorithm to pick the restaurant for lunch over one month (see HW4), Alice is unsatisfied by its performance and feeling a lot of "regret". She suspects that this is because the quality of the meals in these restaurants does not follow a fixed distribution. Therefore, she decides to switch to the Exp3 algorithm with learning rate $\eta = 0.1$ to choose among the 3 restaurants that she is interested in, but she needs your help to implement this idea.

6.1 Describe in one sentence how Alice should pick the restaurant on the first day. (1 points)

She should pick the three restaurants uniformly at random.

6.2 Suppose that Alice ends up picking the first restaurant on the first day and really likes it (that is, reward is 1). Describe how Alice should pick the restaurant on the second day and explain why. (3 points)

Alice should still pick the restaurant uniformly at random on the second day. This is because we should use losses instead of rewards in Exp3 to ensure proper exploration. In other words, we should first convert the reward of 1 for the first restaurant to a loss of 0, which then leads to an estimated loss of 0 for all 3 restaurants. After applying softmax, this would result in a uniform distribution again.

Rubrics: 1 point for the correct answer, 1 point for mentioning the exploration issue regarding "losses" versus "rewards", and 1 point for explaining why this results in a uniform distribution again.

6.3 Suppose that Alice ends up picking the second restaurant on the second day and really dislikes it (that is, reward is 0). Describe how Alice should pick the restaurant on the third day and explain why. (6 points)

Again, we first convert the reward of 0 to a loss of 1. (1 point)

Since the probability of picking the second restaurant on the second day was $1/3$, the importance weighted loss estimator for the three restaurants should be $0, 3$, and 0 respectively. (2 points)

Multiplying this with $-\eta = -0.1$ and further applying softmax gives the final distribution

$$\left(\frac{e^0}{e^0 + e^{-0.3} + e^0}, \frac{e^{-0.3}}{e^0 + e^{-0.3} + e^0}, \frac{e^0}{e^0 + e^{-0.3} + e^0} \right),$$

which is what Alice should sample the restaurant from on the third day. (3 points)

You can use this page for scratch work

You can use this page for scratch work

You can use this page for scratch work

You can use this page for scratch work

You can use this page for scratch work