CSCI-567 Fall 2019 Midterm Exam 2 Ans: [Rubric]

Problem	1	2	3	4	5	Total
Points	20	25	20	20	15	100

Please read the following instructions carefully:

- The exam has a total of **14 pages** (including this cover and two blank pages in the end). Each problem have several questions. Once you are permitted to open your exam (and not before), you should check and make sure that you are not missing any pages.
- Duration of the exam is **2 hours and 20 mins**. Questions are not ordered by their difficulty. Budget your time on each question carefully.
- Select **one and only one answer** for all multiple choice questions.
- Answers should be **concise** and written down **legibly**. All questions can be done within 5-12 lines.
- You must answer each question on the page provided. You can use the last two blank pages as scratch paper. Raise your hand to ask a proctor for more if needed.
- This is a **closed-book/notes** exam. Consulting any resources is NOT permitted.
- Any kind of cheating will lead to **score 0** for the entire exam and be reported to SJACS.

1 Multiple Choice, True or False

1. A decision stump can only lead to linear decision boundary for classification. Ans: A
   (a) True.
   (b) False.

2. The AdaBoost algorithm will eventually reach zero training error regardless of the type of weak classifier it uses, when enough iterations are performed. Ans: B
   (a) True.
   (b) False.

3. Which of the following statement is true? Ans: D
   (A) In the Adaboost algorithm, weights of the misclassified examples may not go up.
   (B) Boosting algorithm cannot select the same weak classifier more than once.
   (C) The testing error of the classifier learned with Adaboost algorithm (combination of all the weak classifier) monotonically increases as the number of iterations in the boosting algorithm increases.
   (D) None of the above

4. When applying a GMM of K components to a dataset of N points, if we representing γ_{nk} (which is the term used to update component weights) as a matrix of N rows and K columns, what is the sum of this matrix? Ans: C
   (A) 1
   (B) K
   (C) N
   (D) NK

5. How many learnable parameters are there in a GMM with K components and full covariance when the GMM is applied to a dataset of N points, each being D dimensional? Ans: B
   (A) $K(D + D(D - 1))$
   (B) $KD(D + 1)$
   (C) $K(N + D^2)$
   (D) $KD + NK^2$

6. Which of the following models has a continuous latent variable? Ans: B
   (a) Naive Bayes Classifier
   (b) Principal Component Analysis
   (c) Gaussian Mixture Model
   (d) Hidden Markov Model

7. Given the parameters of an HMM and an observation sequence O, we can determine the likelihood $P(O \mid \lambda)$. NOTE: λ represents the parameters of the HMM model. Ans: A

(a) True

(b) False

8. Given an observation sequence O and the set of possible states in the HMM, we can learn the HMM parameters, including transition probability matrix and emission probabilities. Ans: A

   (A) True

   (B) False

9. Which is TRUE about the Baum-Welsh algorithm? Ans: B

   (A) It is used to find the real parameters of a hidden markov model.

   (B) It uses a forward-backward algorithm to maximize the probability of an observation.

   (C) The forward-backward algorithm can not do completely unsupervised learning of the transition matrix A and emission matrix B parameters.

   (D) It is a special case of the EM algorithm which is a recursive algorithm.

10. Which of the following statements is NOT TRUE about Lagrangian duality? Ans: A

    (A) Optimal values of the primal and dual problems need to be equal.

    (B) Duality gives us an option of trying to solve our original (potentially nonconvex) constrained optimization problem in another way.

    (C) Duality allows us to formulate optimality conditions for constrained optimization problems.

    (D) One purpose of Lagrange duality is to find a lower bound on a minimization problem or an upper bounds for a maximization problem.

11. What do we mean by generalization error in terms of the SVM? Ans: B

    (A) How far the hyperplane is from the support vectors

    (B) How accurately the SVM can predict outcomes for unseen data

    (C) The threshold amount of error in an SVM

    (D) None of the above.

12. Which of the following statements is NOT TRUE about the differences between SVM and logistic regression (LR)? Ans: A

    (A) Both LR and SVM can give us an unconstrained, smooth objective.

    (B) SVMs have a nice dual form, giving sparse solutions when using the kernel trick.

    (C) They use different loss functions.

    (D) SVM is better at generalization since it doesn't penalize examples for which the correct decision is made with sufficient confidence.

2 Mixture Models (25 points)

Consider a Poisson Mixture Model with the following probability mass function:

$$p(x_n) = \sum_{k=1}^{K} p(x_n, z = k) = \sum_{k=1}^{K} p(z = k)p(x_n \mid z = k) = \sum_{k=1}^{K} \omega_k \pi(x_n \mid \lambda_k) = \sum_{k=1}^{K} \omega_k \frac{\lambda^{x_n}}{x_n!} e^{-\lambda_k}$$

where ω_k is the mixture weight such that $\sum_{k=1}^{K} \omega_k = 1$, x_n is a non-negative integer, K is the number of mixtures, and $\lambda_k > 0$ is the parameter of a Poisson distribution.

Similar to Gaussian Mixture Models, The expected complete log-likelihood is defined as

$$\mathcal{Q} = \sum_{n=1}^{N} \sum_{k=1}^{K} \big(\gamma_{nk} \log p(x_n, z = k) - \gamma_{nk} \log \gamma_{nk}\big),$$

where N is the number of data points.

13. To get the optimal ω_k, we have the following optimization problem:

$$\begin{gathered} \arg_{\omega_k} \max \mathcal{Q}, \\ s.t.\ \omega_k \geq 0, \\ \sum_{k=1}^{K} \omega_k = 1. \end{gathered}$$

Write out the Lagrangian and find the optimal ω_k (treating all other variables constant). **(15 points)**

Ans:

$$\begin{aligned} \arg_{\omega_k} \max \mathcal{Q} &= \arg_{\omega_k} \max \sum_{n=1}^{N} \sum_{k=1}^{K} \big(\gamma_{nk} \log p(x_n, z = k) - \gamma_{nk} \log \gamma_{nk}\big) \quad \text{(definition)} \\ &= \arg_{\omega_k} \max \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} \log p(x_n, z = k) \quad (\gamma_{nk} \text{ is unrelated to } \omega_k) \\ &= \arg_{\omega_k} \max \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} \log \omega_k \frac{\lambda^{x_n}}{x_n!} e^{-\lambda_k} \quad \text{(definition)} \\ &= \arg_{\omega_k} \max \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} \log \omega_k \quad (\text{remove terms unrelated to } \omega_k) \end{aligned}$$

Excluding the terms unrelated to ω_k, we can write the Lagrangian as

$$\mathcal{L} = \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} \log \omega_k + \mu(\sum_{k} \omega_k - 1) + \sum_{k} \mu_k \omega_k \textbf{(5 points)}$$

Set the derivative to zero:

$$\nabla_{\omega_k} \mathcal{L} = \sum_{n} \gamma_{nk} \frac{1}{\omega_k} + \mu + \mu_k = 0. \textbf{(7 points)} \tag{1}$$

By complementary slackness, we have $\mu_k = 0$ 1 and (1) becomes

$$\frac{1}{\omega_k}\sum_n \gamma_{nk} = -\mu \Rightarrow \omega_k = -\frac{1}{\mu}\sum_n \gamma_{nk} \textbf{(9 points)} \tag{2}$$

By feasibility, we have

$$\begin{aligned}\sum_k \omega_k = 1 &\Rightarrow \sum_k -\frac{\sum_n \gamma_{nk}}{\mu} = 1 \\ &\Rightarrow \mu = -\sum_n \sum_k \gamma_{nk} = N \textbf{(11 points)}\end{aligned}$$

Thus, (2) becomes

$$\begin{aligned}\omega_k &= \frac{\sum_n \gamma_{nk}}{\sum_n \sum_k \gamma_{nk}} \textbf{(13 points)} \\ &= \frac{1}{N}\sum_n \gamma_{nk} \textbf{(15 points)}\end{aligned}$$

Remarks:

- x partial credits: only the first few steps are correct.
- incorrect answer: everything before the final answer is correct.
- incomplete answer: mostly correct, with small defects such as not representing the denominator with N.

14. Find the optimal λ_k (treating all other variables constant). **(10 points)**

Ans: Set the derivative to zero, we have

$$\log p(x_n, z = k) = \log \omega_k \frac{\lambda_k^{x_n}}{x_n!} e^{-\lambda_k} \tag{3}$$
$$= \log \omega_k + x_n \log \lambda_k - \log x_n! - \lambda_k \tag{4}$$

$$\begin{aligned}
\arg_{\lambda_k} \max \mathcal{Q} &= \arg_{\lambda_k} \max \sum_{n=1}^{N} \sum_{k=1}^{K} \left(\gamma_{nk} \log p(x_n, z = k) - \gamma_{nk} \log \gamma_{nk}\right) \quad \text{(definition)} \\
&= \arg_{\lambda_k} \max \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} \log p(x_n, z = k) \quad (\gamma_{nk} \text{ is unrelated to } \lambda_k) \\
&= \arg_{\lambda_k} \max \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} (\log \omega_k + x_n \log \lambda_k - \log x_n! - \lambda_k) \quad \text{(from (3))} \\
&= \arg_{\lambda_k} \max \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} (x_n \log \lambda_k - \lambda_k) \quad \text{(remove terms unrelated to } \lambda_k) \\
&= \arg_{\lambda_k} \max \mathcal{Q}^{'}
\end{aligned}$$

$$\begin{aligned}
\nabla_{\lambda_k} \mathcal{Q} &= \nabla_{\lambda_k} \mathcal{Q}^{'} \\
&= \sum_{n=1}^{N} \nabla_{\lambda_k} \sum_{k=1}^{K} \gamma_{nk} (x_n \log \lambda_k - \lambda_k) \\
&= \sum_{n} \gamma_{nk} (\frac{1}{\lambda_k} x_n - 1) = 0 \\
\Rightarrow \frac{1}{\lambda_k} \sum_{n} \gamma_{nk} x_n &= \sum_{n} \gamma_{nk} \\
\Rightarrow \lambda_k &= \frac{\sum_n \gamma_{nk} x_n}{\sum_n \gamma_{nk}}
\end{aligned}$$

Remarks:

- basic points: only the Langrage is correct.
- x partial credits: only the first few steps are correct.
- wrong expansion/derivation: the derivative is wrong.

3 Support Vector Machine (20 points)

Given an unlabeled set of examples $\{x_1, \ldots, x_N\}$, the one-class SVM algorithm tries to find a direction $\mathbf{w}$ that maximally separates the data from the origin. More precisely, it solves the following optimization problem:

$$\min_{\mathbf{w}} \frac{1}{2}\mathbf{w}^T\mathbf{w}$$
$$s.t. \quad \mathbf{w}^T\mathbf{x}_n \geq 1 \ \ \forall n = \{1, \ldots, N\}$$

A new test example x is labeled 1 if $\mathbf{w}^T\mathbf{x} \geq 1$, and 0 otherwise.

15. Write down the corresponding dual optimization problem for the above. Your answer should not have any term of $\mathbf{w}$. **(10 points)**

Ans:

$$L\left(\mathbf{w}, \lambda_i\right) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + \sum_{i=1}^{N} \lambda_i \left(1 - \mathbf{w}^T\mathbf{x}_i\right) \textbf{ (2 points)}$$

$$\frac{dL}{d\mathbf{w}} = \mathbf{w} - \sum_{i=1}^{N} \lambda_i \mathbf{x}_i = 0 \textbf{(2 points)}$$

$$\mathbf{w} = \sum_{i=1}^{N} \lambda_i \mathbf{x}_i \textbf{(2 points)}$$

Substituting we get,

$$\tilde{L}(\lambda_i) = \sum_{i=1}^{N} \lambda_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i \lambda_j \mathbf{x}_i^T \mathbf{x}_j \textbf{(2 points)}$$

and dual objective would be

$$\max_{\lambda_i, \lambda_i \geq 0} \tilde{L}(\lambda_i)$$

(2 points)

16. Can the one-class SVM be kernelised in training? How? **(4 points)**

Ans: Since, the dual is expressed only in terms of dot product, it can be kernelized easily **(2 points)**
and dual objective would become

$$\tilde{L}(\lambda_i) = \sum_{i=1}^{N} \lambda_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i \lambda_j k(\mathbf{x}_i, \mathbf{x}_j) \textbf{(2 points)}$$

17. Can the one-class SVM be kernelised in testing? How? **(6 points)**

Ans: Note, that for testing we need to check value of $\mathbf{w}^T\mathbf{x}$, **(2 points)**
which is $\sum_{i=1}^{N} \lambda_i \mathbf{x}_i^T \mathbf{x} = \sum_{i=1}^{N} \lambda_i k(\mathbf{x}_i, \mathbf{x})$ **(2 points)**
Hence, it can be kernelized in testing too **(2 points)**

4 Boosting (20 points)

In this question we will look into the AdaBoost algorithm (shown in Alg. 1), where the base algorithm is simply searching for a classifier with the smallest weighted error from a fixed classifier set $\mathcal{H}$.

Algorithm 1: Adaboost

Given: A training set $\{(\boldsymbol{x}_n, y_n \in \{+1, -1\})\}_{n=1}^{N}$, and a set of classifier $\mathcal{H}$, where each $h \in \mathcal{H}$ takes a feature vector as input and outputs $+1$ or -1.

Goal: Learn $H(\boldsymbol{x}) = \operatorname{sign}\left(\sum_{t=1}^{T} \beta_t h_t(\boldsymbol{x})\right)$, where $h_t \in \mathcal{H}$, $\beta_t \in \mathbb{R}$, and $\operatorname{sign}(a) = \begin{cases} +1, & \text{if } a \geq 0, \\ -1, & \text{otherwise.} \end{cases}$

Initialization: $D_1(n) = \frac{1}{N}$, $\forall n \in [N]$.

for $t = 1, 2, \cdots, T$ **do**

  Find $h_t = \arg\min_{h \in \mathcal{H}} \sum_{n: y_n \neq h(\boldsymbol{x}_n)} D_t(n)$.

  Compute

$$\epsilon_t = \sum_{n: y_n \neq h_t(\boldsymbol{x}_n)} D_t(n) \qquad \text{and} \qquad \beta_t = \frac{1}{2} \log \frac{1 - \epsilon_t}{\epsilon_t}.$$

  Compute

$$D_{t+1}(n) = \frac{D_t(n) e^{-\beta_t y_n h_t(\boldsymbol{x}_n)}}{\sum_{n'=1}^{N} D_t(n') e^{-\beta_t y_{n'} h_t(\boldsymbol{x}_{n'})}}$$

  for each $n \in [N]$

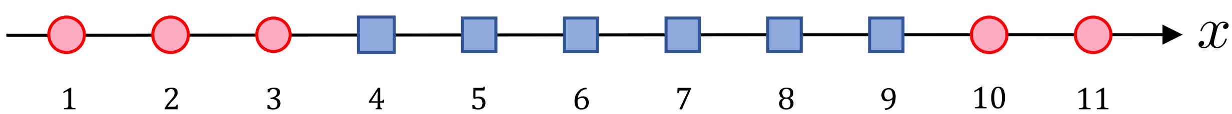

Figure 1: The 1-dimensional training set with 11 data. A square means the class of the data is $+1$, *i.e.* $y = +1$ and a circle means $y = -1$. The number under each data indicates its x coordinate.

Now we are given a training set of 11 data as shown in Fig. 1. Each training data is 1-dimension and denoted as a square or a circle in the figure, where the square refers the class of the data is $+1$, *i.e.* $y = +1$ and the circle refers $y = -1$. You are going to experiment on the given training set with the learning process of the AdaBoost algorithm as shown in Alg. 1 for $T = 2$. The base classifier set $\mathcal{H}$ consists of all decision stumps, where each of the decision stumps is parameterized by a pair $(s, b) \in \{+1, -1\} \times \mathbb{R}$ such that

$$h_{(s,b)}(x) = \begin{cases} s, & \text{if } x > b, \\ -s, & \text{otherwise.} \end{cases}$$

Throughout this problem, the natural logarithm which has the e (≈ 2.71828) as its base is applied for the $\log(\cdot)$.

18. Please write down the pair (s, b) of the best decision stump h_1, and ϵ_1 at $t = 1$. If there are multiple equally optimal stump functions, just randomly pick **one** of them to be h_1. Write down which data points are mis-classified, and ϵ_1. ϵ_1 should be in the form of **a fraction**. **(5 points)**

Ans: $s = 1$ and $b \in [3, 4)$ (any b in this range is fine). **(2 points)**
The right-most two circles will be mis-classified. **(1 points)**
$\epsilon_1 = \frac{2}{11}$ **(2 points)**

19. Please write down the pair (s, b) of the best decision stump h_2, and ϵ_2 at $t = 2$. If there are multiple equally best stump functions, just randomly pick **one** of them to be h_2. Write ϵ_2 in the form of **a fraction** following Alg. 1. You don't need to compute the actual value of $e^{\frac{1}{2}\log c}$, instead try to cancel them out with the $\exp(\cdot)$ and the $\log(\cdot)$ properties. **(7 points)**

Ans: $s = -1$ and $b \in [9, 10)$ (any b is this range is fine). **(3 points)**
The left-most 3 circles will be mis-classified.

$$\epsilon_2 = \frac{\frac{3}{11}e^{-\frac{1}{2}\log\frac{9}{2}}}{\frac{2}{11}e^{\frac{1}{2}\log\frac{9}{2}} + \frac{9}{11}e^{-\frac{1}{2}\log\frac{9}{2}}} \tag{5}$$
$$= \frac{3e^{-\frac{1}{2}\log\frac{9}{2}}}{2e^{\frac{1}{2}\log\frac{9}{2}} + 9e^{-\frac{1}{2}\log\frac{9}{2}}} \tag{6}$$
$$= \frac{3}{2e^{\log\frac{9}{2}} + 9} \tag{7}$$
$$= \frac{3}{9+9} \tag{8}$$
$$= \frac{3}{18} = \frac{1}{6} \textbf{ (4 points)} \tag{9}$$

20. Now we have the final classifier $H(\boldsymbol{x}) = \text{sign}(\beta_1 h_1 + \beta_2 h_2)$: write down β_1, β_2, the class predicted by $H(\boldsymbol{x})$ for each data point and the final training accuracy. The training accuracy should be presented as a fraction. **(8 points)**.

Ans:

Substitute $\beta_1 = \frac{1}{2}\log\frac{9}{2}$ and $\beta_2 = \frac{1}{2}\log 5$ **(3 points)** into $H(\boldsymbol{x}) = sign(\beta_1 h_1 + \beta_2 h_2)$, we have that:

$$H(\boldsymbol{x}) = sign(\beta_1 h_1 + \beta_2 h_2) = sign(\frac{1}{2}(\log\frac{9}{2})h_1 + \frac{1}{2}(\log 5)h_2)$$

Data	1	2	3	4	5	6	7	8	9	10	11
Label	+1	+1	+1	+1	+1	+1	+1	+1	+1	-1	-1

The class of each data is listed in the table **(2 points)**.

Training accuracy is $\frac{8}{11}$. **(3 points)**

5 HMM (15 points)

Recall a hidden Markov model is parameterized by

- initial state distribution $P(X_1 = s) = \pi_s$
- transition distribution $P(X_{t+1} = s' | X_t = s) = a_{s,s'}$
- emission distribution $P(O_t = o | X_t = s) = b_{s,o}$

21. Given a sequence of observations $O_{1:t-1}, O_{t+1:T}$, O_t is missing for some reason. Compute the probability of O_t, which is $P(O_t = o | O_{1:t-1}, O_{t+1:T})$ in terms of forward message, backward message, transition probability, emission probability as needed. If $P(O_{1:t-1}, O_{t+1:T})$ appears in the denominator of your solution, you don't need to further express it in other forms. **(10 points)**

$$\alpha_s(t) = P(X_t = s, O_{1:t} = o_{1:t})$$
$$\beta_s(t) = P(O_{t+1:T} = o_{t+1:T} | X_t = s)$$

Ans: solution version 1:

$$\begin{aligned}
P(O_t = o | O_{1:t-1}, O_{t+1:T}) &= \sum_s P(O_t = o, X_t = s | O_{1:t-1}, O_{t+1:T}) \textbf{(1 points)} \\
&= \sum_s P(O_t = o | X_t = s, O_{1:t-1}, O_{t+1:T}) P(X_t = s | O_{1:t-1}, O_{t+1:T}) \textbf{(1 points)} \\
&= \sum_s P(O_t = o | X_t = s) P(X_t = s | O_{1:t-1}, O_{t+1:T}) \textbf{(1 points)} \\
&= \sum_s b_{s,o} P(X_t = s | O_{1:t-1}, O_{t+1:T}) \textbf{(1 points)} \\
&\propto \sum_s b_{s,o} P(X_t = s, O_{1:t-1}, O_{t+1:T}) \textbf{(1 points)} \\
&= \sum_s b_{s,o} P(O_{t+1:T} | X_t = s, O_{1:t-1}) P(X_t = s, O_{1:t-1}) \textbf{(1 points)} \\
&= \sum_s b_{s,o} P(O_{t+1:T} | X_t = s) \sum_{s'} P(X_t = s, X_{t-1} = s', O_{1:t-1}) \textbf{(1 points)} \\
&= \sum_s b_{s,o} \beta_s(t) \sum_{s'} P(X_t = s | X_{t-1} = s', O_{1:t-1}) P(X_{t-1} = s', O_{1:t-1}) \textbf{(1 points)} \\
&= \sum_s b_{s,o} \beta_s(t) \sum_{s'} a(s', s) \alpha_{s'}(t-1) \textbf{(2 points)} \\
or &= \frac{\sum_s b_{s,o} \beta_s(t) \sum_{s'} a(s', s) \alpha_{s'}(t-1)}{P(O_{1:t-1}, O_{t+1:T})} \textbf{(2 points)}
\end{aligned}$$

solution version 2:

$$
\begin{aligned}
P\left(O_t = o | O_{1:t-1}, O_{t+1:T}\right) &\propto P\left(O_t = o, O_{1:t-1}, O_{t+1:T}\right) \textbf{ (1 points)} \\
&= \sum_s P\left(O_{t+1:T} = o, X_t = s, O_{1:t-1}, O_t\right) \textbf{(1 points)} \\
&= \sum_s P\left(O_{t+1:T} = o | X_t = s, O_{1:t-1}, O_t\right) P\left(X_t = s, O_{1:t}\right) \textbf{(1 points)} \\
&= \sum_s P\left(O_{t+1:T} = o | X_t = s\right) P\left(X_t = s, O_{1:t}\right) \textbf{(1 points)} \\
&= \sum_s \beta_s(t)\alpha_s(t) \textbf{(6 points)} \\
or &= \frac{\sum_s \beta_s(t)\alpha_s(t)}{P\left(O_{1:t-1}, O_{t+1:T}\right)} \textbf{(6 points)}
\end{aligned}
$$

solution version 3:

$$
\begin{aligned}
P\left(O_t = o | O_{1:t-1}, O_{t+1:T}\right) &\propto P\left(O_t = o, O_{1:t-1}, O_{t+1:T}\right) \textbf{ (1 points)} \\
&= \sum_s P\left(X_t = s, O_{1:T}\right) \textbf{(1 points)} \\
&= \sum_s \alpha_s(T) \textbf{(8 points)} \\
or &= \frac{\sum_s \alpha_s(T)}{P\left(O_{1:t-1}, O_{t+1:T}\right)} \textbf{(8 points)}
\end{aligned}
$$

solution version 4:

$$
\begin{aligned}
P\left(O_t = o | O_{1:t-1}, O_{t+1:T}\right) &\propto P\left(O_t = o, O_{1:t-1}, O_{t+1:T}\right) \textbf{ (1 points)} \\
&= \sum_s P\left(X_t = s, O_{1:T}\right) \textbf{(1 points)} \\
&= \sum_s P(X_t = s, O_{1:t}) P(O_{t+1:T} | X_t = s, O_{1:t}) \textbf{(1 points)} \\
&= \sum_s \alpha_s(t) P(O_{t+1:T} | X_t = s) \textbf{(1 points)} \\
&= \sum_s \alpha_s(t) P(O_{t+1:T}, X_t = s)/P(X_t = s) \textbf{(1 points)} \\
&= \sum_s \alpha_s(t) \sum_{s'} P(O_{t+2:T}, O_{t+1}, X_{t+1} = s', X_t = s)/P(X_t = s) \textbf{(1 points)} \\
&= \sum_s \alpha_s(t) \sum_{s'} P(X_{t+1} = s' | X_t = s) P(O_{t+1} | X_{t+1} = s', X_t = s) P(O_{t+2:T} | O_{t+1}, X_{t+1} = s', X_t = s)(\\
&= \sum_s \alpha_s(t) \sum_{s'} a_{s,s'} b_{s',o_{t+1}} \beta_{s'}(t+1) \textbf{(3 points)} \\
or &= \frac{\sum_s \alpha_s(t) \sum_{s'} a_{s,s'} b_{s',o_{t+1}} \beta_{s'}(t+1)}{P\left(O_{1:t-1}, O_{t+1:T}\right)} \textbf{(3 points)}
\end{aligned}
$$

Since $\sum_s \alpha_s(T) = \sum_s \alpha_s(t)\beta_s(t)$, and $\alpha_s(t) = \sum_{s'} b_{s,o} a_{s',s} \alpha_{s'}(t-1)$, and $\beta_s(t) = \sum_{s'} b_{s',o_{t+1}} a_{s,s'} \beta_{s'}(t+1)$, all the above solutions are equal.

22. Consider a HMM model with states $X_t \in \{S_1, S_2, S_3\}$, observations $O_t \in \{A, B, C\}$, and parameters

$$\pi_1 = P(X_1 = S_1) = 0; \quad \pi_2 = P(X_1 = S_2) = 0; \quad \pi_3 = P(X_1 = S_3) = 1$$

$$\begin{aligned} &a_{11} = 0, \quad a_{12} = 0, \quad a_{13} = 1 \\ &a_{21} = 1/2, \quad a_{22} = 1/2, \quad a_{23} = 0 \\ &a_{31} = 1/3, \quad a_{32} = 1/3, \quad a_{33} = 1/3 \end{aligned}$$

$$\begin{aligned} &b_1(A) = 0, \quad b_1(B) = 1/2, \quad b_1(C) = 1/2 \\ &b_2(A) = 1/2, \quad b_2(B) = 0, \quad b_2(C) = 1/2 \\ &b_3(A) = 1/2, \quad b_3(B) = 1/2, \quad b_3(C) = 0 \end{aligned}$$

What is $P(X_3 = S_1)$? hint: you do not need to run Viterbi algorithm. **(5 points)**

Ans:

$$\begin{aligned} P(X_3 = S_2) &= 1/3 \times 1/3 + 1/3 \times 1/2 = 5/18 \\ P(X_3 = S_3) &= 1/3 \times 1/3 + 1/3 \times 1 = 4/9 \\ 1 - P(X_3 = S_2) - P(X_3 = S_3) &= 1 - 5/18 - 4/9 = 5/18 \textbf{(5 points)} \end{aligned}$$

You may use this page as scratch paper.

You may use this page as scratch paper.