10-315 Machine Learning	**Name:**
Spring 2023	**Andrew Email:**
Exam 2 Practice Problems	**Room:**
April 25, 2023	**Seat:**
Time Limit: N/A	**Exam Number:**

Instructions:

- Fill in your name and Andrew ID above. Be sure to write neatly, or you may not receive credit for your exam.
- Clearly mark your answers in the allocated space **on the front of each page.** If needed, use the back of a page for scratch space, but you will not get credit for anything written on the back of a page. If you have made a mistake, cross out the invalid parts of your solution, and circle the ones which should be graded.
- No electronic devices may be used during the exam.
- Please write all answers in pen.
- You have N/A to complete the exam. Good luck!

Instructions for Specific Problem Types

For "Select One" questions, please fill in the appropriate bubble completely:

Select One: Who taught this course?

● Pat Virtue

○ Marie Curie

○ Noam Chomsky

If you need to change your answer, you may cross out the previous answer and bubble in the new answer:

Select One: Who taught this course?

● Pat Virtue

○ Marie Curie

⊘ Noam Chomsky

For "Select all that apply" questions, please fill in all appropriate squares completely:

Select all that apply: Which are scientists?

■ Stephen Hawking

■ Albert Einstein

■ Isaac Newton

☐ I don't know

Again, if you need to change your answer, you may cross out the previous answer(s) and bubble in the new answer(s):

Select all that apply: Which are scientists?

■ Stephen Hawking

■ Albert Einstein

■ Isaac Newton

⊘ I don't know

For questions where you must fill in a blank, please make sure your final answer is fully included in the given space. You may cross out answers or parts of answers, but the final answer must still be within the given space.

Fill in the blank: What is the course number?

10-315

10-~~7~~315

1 Optimization

1. **Select all that apply:** Which of the following are correct regarding Gradient Descent (GD) and stochastic gradient descent (SGD)

   - ☐ Each update step in SGD pushes the parameter vector closer to the parameter vector that minimizes the objective function.
   - ☐ The gradient computed in SGD is, in expectation, equal to the gradient computed in GD.
   - ☐ The gradient computed in GD has a higher variance than that computed in SGD, which is why in practice SGD converges faster in time than GD.

2. (a) Determine if the following 1-D functions are convex. Assume that the domain of each function is $\mathbb{R}$. The definition of a convex function is as follows:

   $f(x)$ is convex $\iff f(\alpha x + (1-\alpha)z) \leq \alpha f(x) + (1-\alpha)f(z), \forall \alpha \in [0,1]$ and $\forall x, z$.

   Select all convex functions:

   - ☐ $f(x) = x + b$ for any $b \in \mathbb{R}$
   - ☐ $f(x) = c^2 x$ for any $c \in \mathbb{R}$
   - ☐ $f(x) = ax^2 + b$ for any $a \in \mathbb{R}$ and any $b \in \mathbb{R}$
   - ☐ $f(x) = 0$
   - ☐ None of the Above

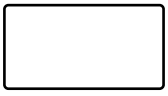
   (b) Consider the convex function $f(z) = z^2$. Let α be our learning rate in gradient descent.

   For which values of α will $\lim_{t\to\infty} f(z^{(t)}) = 0$, assuming the initial value of z is $z^{(0)} = 1$ and $z^{(t)}$ is the value of z after the t-th iteration of gradient descent.

   Select all that apply:

   - ☐ $\alpha = 0$
   - ☐ $\alpha = \frac{1}{2}$
   - ☐ $\alpha = 1$
   - ☐ $\alpha = 2$
   - ☐ None of the Above

   Give the range of all values for $\alpha \geq 0$ such that $\lim_{t\to\infty} f(z^{(t)}) = 0$, assuming the initial value of z is $z^{(0)} = 1$. Be specific.



2 Logistic Regression

1. **[2 pts]** If today I want to predict the probability that a student sleep more than 8 hours on average (SA) given the Course loading (C), I will choose to use linear regression over logistic regression.

   Circle one: True False

2. Answer the following questions with brief explanations where necessary.

   a) **[2 pts]** A generalization of logistic regression to a multiclass settings involves expressing the per-class probabilities $P(y = c|x)$ as the softmax function $\frac{\exp(w_c^T x)}{\sum_{d \in C} \exp(w_d^T x)}$, where c is some class from the set of all classes C.

   Consider a 2-class problem (labels 0 or 1). Rewrite the above expression for this situation, to end up with expressions for $P(Y = 1|x)$ and $P(Y = 0|x)$ that we have already come across in class for binary logistic regression.

   b) **[3 pts]** Given 3 data points $(1, 1), (1, 0), (0, 0)$ with labels $0, 1, 0$ respectively. Consider 2 models, Model 1: $\sigma(w_1 x_1 + w_2 x_2)$, Model 2: $\sigma(w_0 + w_1 x_1 + w_2 x_2)$ ($\sigma(z)$ is the sigmoid function $\frac{1}{1+e^{-z}}$) that compute $p(y = 1|\mathbf{x})$. Using the given data, we can learn parameters $\hat{w}$ by maximizing the conditional log-likelihood.

   Suppose we switched $(0, 0)$ to label 1 instead.

   Do the parameters learnt for Model 1 change?

   Circle one: True False
   One-line explanation:

   What about Model 2?

   Circle one: True False
   One-line explanation:

   c) **[2 pts]** For logistic regression, we need to resort to iterative methods such as gradient descent to compute the $\hat{w}$ that maximizes the conditional log likelihood. Why?

   d) **[3 pts]** Considering a Gaussian prior, write out the MAP objective function $J(w)_{MAP}$ in terms of the MLE objective $J(w)_{MLE}$. Name the variant of logistic regression this results in.

3. Given a training set $\{(x_i, y_i), i = 1, \ldots, n\}$ where $x_i \in \mathbb{R}^d$ is a feature vector and $y_i \in \{0, 1\}$ is a binary label, we want to find the parameters $\hat{w}$ that maximize the likelihood for the training set, assuming a parametric model of the form

$$p(y = 1|x; w) = \frac{1}{1 + \exp(-w^T x)}.$$

The conditional log likelihood of the training set is

$$\ell(w) = \sum_{i=1}^{n} y_i \log p(y_i, |x_i; w) + (1 - y_i) \log(1 - p(y_i, |x_i; w)),$$

and the gradient is

$$\nabla \ell(w) = \sum_{i=1}^{n} (y_i - p(y_i|x_i; w))x_i.$$

a) [5 pts.] Is it possible to get a closed form for the parameters $\hat{w}$ that maximize the conditional log likelihood? How would you compute $\hat{w}$ in practice?

b) [5 pts.] For a binary logistic regression model, we predict $y = 1$, when $p(y = 1|x) \geq 0.5$. Show that this is a linear classifier.

c) Consider the case with binary features, i.e, $x \in \{0, 1\}^d \subset \mathbb{R}^d$, where feature x_1 is rare and happens to appear in the training set with only label 1. What is $\hat{w}_1$? Is the gradient ever zero for any finite w? Why is it important to include a regularization term to control the norm of $\hat{w}$?

4. Given the following dataset, $\mathcal{D}$, and a fixed parameter vector, $\boldsymbol{\theta}$, write an expression for the binary logistic regression conditional likelihood.

$\mathcal{D} = \{(\mathbf{x}^{(1)}, y^{(1)} = 0), (\mathbf{x}^{(2)}, y^{(2)} = 0), (\mathbf{x}^{(3)}, y^{(3)} = 1), (\mathbf{x}^{(4)}, y^{(4)} = 1)\}$

- Write your answer in terms of $\boldsymbol{\theta}$, $\mathbf{x}^{(1)}$, $\mathbf{x}^{(2)}$, $\mathbf{x}^{(3)}$, and $\mathbf{x}^{(4)}$.
- Do not include $y^{(1)}$, $y^{(2)}$, $y^{(3)}$, or $y^{(4)}$ in your answer.
- Don't try to simplify your expression.

Conditional likelihood:

5. Write an expression for the decision boundary of binary logistic regression with a bias term for two-dimensional input features $x_1 \in \mathbf{R}$ and $x_2 \in \mathbf{R}$ and parameters b (the intercept parameter), w_1, and w_2. Assume that the decision boundary occurs when $P(Y = 1 \mid \mathbf{x}, b, w_1, w_2) = P(Y = 0 \mid \mathbf{x}, b, w_1, w_2)$.

(a) Write your answer in terms of x_1, x_2, b, w_1, and w_2.

Decision boundary equation:

(b) What is the geometric shape defined by this equation?

6. We have now feature engineered the two-dimensional input, $x_1 \in \mathbb{R}$ and $x_2 \in \mathbb{R}$, mapping it to a new input vector: $\mathbf{x} = \begin{bmatrix} 1 \\ {x_1}^2 \\ {x_2}^2 \end{bmatrix}$

   (a) Write an expression for the decision boundary of binary logistic regression with this feature vector $\mathbf{x}$ and the corresponding parameter vector $\boldsymbol{\theta} = [b, w_1, w_2]^T$. Assume that the decision boundary occurs when $P(Y = 1 \mid x, \boldsymbol{\theta}) = P(Y = 0 \mid x, \boldsymbol{\theta})$. Write your answer in terms of x_1, x_2, b, w_1, and w_2.

   Decision boundary expression:

   (b) What is the geometric shape defined by this equation?

   (c) If we add an L2 regularization on $[w_1, w_2]^T$, what happens to **parameters** as we increase the λ that scales this regularization term?

   (d) If we add an L2 regularization on $[w_1, w_2]^T$, what happens to the **decision boundary shape** as we increase the λ that scales this regularization term?

3 Feature Engineering and Regularization

1. **Model Complexity:** In this question we will consider the effect of increasing the model complexity, while keeping the size of the training set fixed. To be concrete, consider a classification task on the real line $\mathbb{R}$ with distribution D and target function $c^* : \mathbb{R} \to \{\pm 1\}$ and suppose we have a random sample S of size n drawn iid from D. For each degree d, let ϕ_d be the feature map given by $\phi_d(x) = (1, x, x^2, \ldots, x^d)$ that maps points on the real line to $(d+1)$-dimensional space.

   Now consider the learning algorithm that first applies the feature map ϕ_d to all the training examples and then runs logistic regression as in the previous question. A new example is classified by first applying the feature map ϕ_d and then using the learned classifier.

   a) [4 pts.] For a given dataset S, is it possible for the training error to increase when we increase the degree d of the feature map? **Please explain your answer in 1 to 2 sentences.**

   b) [4 pts.] Briefly **explain in 1 to 2 sentences** why the true error first drops and then increases as we increase the degree d.

4 Neural Networks

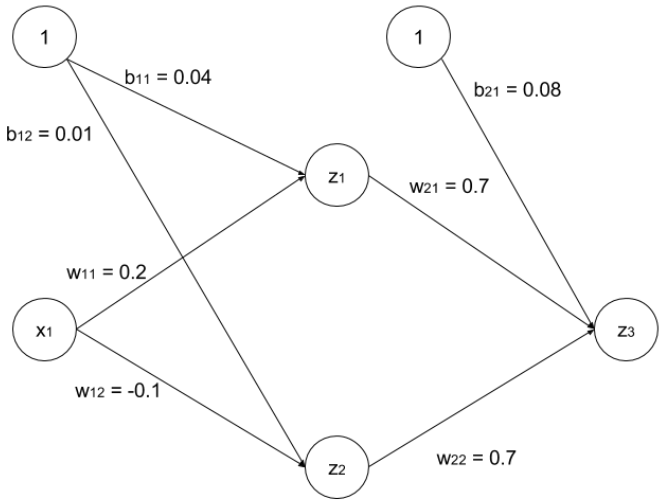


Figure 1: neural network

1. Consider the neural network architecture shown above for a 2-class $(0, 1)$ classification problem. The values for weights and biases are shown in the figure. We define:

$a_1 = w_{11}x_1 + b_{11}$

$a_2 = w_{12}x_1 + b_{12}$

$a_3 = w_{21}z_1 + w_{22}z_2 + b_{21}$

$z_1 = \text{relu}(a_1)$

$z_2 = \text{relu}(a_2)$

$z_3 = \sigma(a_3)$, $\sigma(x) = \frac{1}{1+e^{-x}}$

Use this information to answer the questions that follow.

(i) **[6 pts]** For $x_1 = 0.3$, compute z_3, in terms of e. **Show all work.**
$z_3 =$

(ii) **[2 pts]** To which class does the network predict the given data point ($x_1 = 0.3$), i.e., $\hat{y} =$? Note that $\hat{y} = 1$ if $z_3 > \frac{1}{2}$, else $\hat{y} = 0$.

Circle one: 0 1

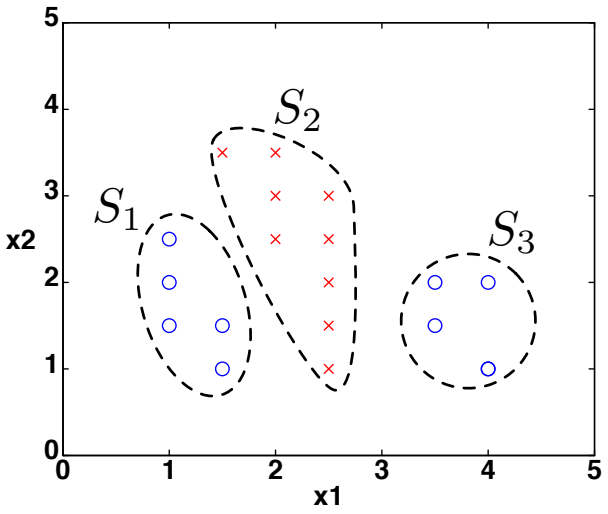
(iii) **[6 pts]** Perform backpropagation on the bias b_{21} by deriving the expression for the gradient of the loss function $L(y, z_3)$ with respect to the bias term b_{21}, $\frac{\partial L}{\partial b_{21}}$, in

terms of the partial derivatives $\frac{\partial \alpha}{\partial \beta}$, where α and β can be any of $L, z_i, a_i, b_{ij}, w_{ij}, x_1$ for all valid values of i, j. Your backpropagation algorithm should be as explicit as possible—that is, make sure each partial derivative $\frac{\partial \alpha}{\partial \beta}$ cannot be decomposed further into simpler partial derivatives. Do *not* evaluate the partial derivatives.

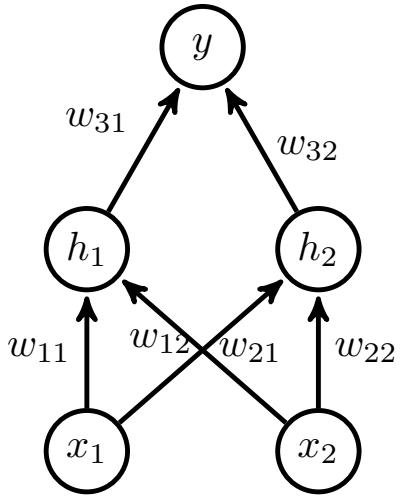
(iv) **[6 pts]** Perform backpropagation on the bias b_{12} by deriving the expression for the gradient of the loss function $L(y, z_3)$ with respect to the bias term b_{12}, $\frac{\partial L}{\partial b_{12}}$, in terms of the partial derivatives $\frac{\partial \alpha}{\partial \beta}$, where α and β can be any of $L, z_i, a_i, b_{ij}, w_{ij}, x_1$ for all valid values of i, j. Your backpropagation algorithm should be as explicit as possible—that is, make sure each partial derivative $\frac{\partial \alpha}{\partial \beta}$ cannot be decomposed further into simpler partial derivatives. Do *not* evaluate the partial derivatives.

2. In this problem we will use a neural network to classify the crosses ($\times$) from the circles ($\circ$) in the simple dataset shown in Figure 2a. Even though the crosses and circles are not linearly separable, we can break the examples into three groups, S_1, S_2, and S_3 (shown in Figure 2a) so that S_1 is linearly separable from S_2 and S_2 is linearly separable from S_3. We will exploit this fact to design weights for the neural network shown in Figure 2b in order to correctly classify this training set. For all nodes, we will use the threshold activation function

$$\phi(z) = \begin{cases} 1 & z > 0 \\ 0 & z \leq 0. \end{cases}$$

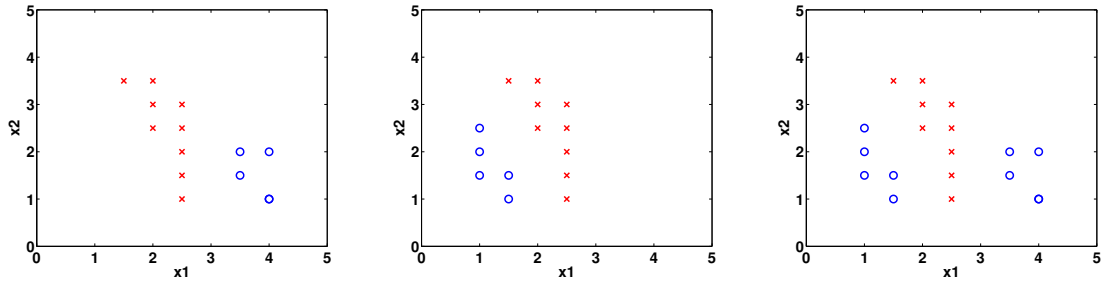


(a) The dataset with groups S_1, S_2, and S_3.



(b) The neural network architecture

Figure 2



(a) Set S2 and S3 (b) Set S1 and S2 (c) Set S1, S2 and S3

Figure 3: NN classification.

(i) First we will set the parameters w_{11}, w_{12} and b_1 of the neuron labeled h_1 so that its output $h_1(x) = \phi(w_{11}x_1 + w_{12}x_2 + b_1)$ forms a linear separator between the sets S_2 and S_3.

(a) [1 pt.] On Fig 3a, draw a linear decision boundary that separates S_2 and S_3.

(b) [1 pt.] Write down the corresponding weights w_{11}, w_{12}, and b_1 so that $h_1(x) = 0$ for all points in S_3 and $h_1(x) = 1$ for all points in S_2. One solution would suffice and the same applies to (ii) and (iii).

(ii) Next we set the parameters w_{21}, w_{22} and b_2 of the neuron labeled h_2 so that its output $h_2(x) = \phi(w_{21}x_1 + w_{22}x_2 + b_2)$ forms a linear separator between the sets S_1 and S_2.

(a) [1 pt.] On Fig 3b, draw a linear decision boundary that separates S_1 and S_2.

(b) [1 pt.] Write down the corresponding weights w_{21}, w_{22}, and b_2 so that $h_2(x) = 0$ for all points in S_1 and $h_2(x) = 1$ for all points in S_2.

(iii) Now we have two classifiers h_1 (to classify S_2 from S_3) and h_2 (to classify S_1 from S_2). We will set the weights of the final neuron of the neural network based on the results from h_1 and h_2 to classify the crosses from the circles. Let $h_3(x) = \phi\big(w_{31}h_1(x) + w_{32}h_2(x) + b_3\big)$.

(a) [1 pt.] Compute w_{31}, w_{32}, b_3 such that $h_3(x)$ correctly classifies the entire dataset.

(b) [1 pt.] Draw your decision boundary in Fig 3c.

(iv) **Back propagation**

In the above example, we need to learn the weights by according to the data. At first step, we need to get the gradients of the parameters of neural networks.

Suppose there m data points x_i with label y_i, where $i \in [1, m]$. x_i is a $d \times 1$ vector and $y_i \in \{0, 1\}$. We use the data to train a neural network with one hidden layer:

$$\begin{aligned} h(x) =& \sigma(W_1 x + b_1) \\ p(x) =& \sigma(W_2 h(x) + b_2), \end{aligned}$$

where $\sigma(x) = \frac{1}{1+\exp(-x)}$ is the sigmoid function, W_1 is a n by d matrix and b_1 is a n by 1 vector, W_2 is a 1 by n matrix and b_1 is a 1 by 1 vector.

We use cross entropy loss function and minimize the negative log likelihood to train the neural network:

$$l = \frac{1}{m}\sum_i l_i = \frac{1}{m}\sum_i -(y_i \log p_i + (1-y_i)\log(1-p_i)),$$

where $p_i = p(x_i), h_i = h(x_i)$.

(a) Describe how you would drive the gradients w.r.t the parameters W_1, W_2 and b_1, b_2. (No need to write out the detailed mathematical expression.)

(b) When m is large, we typically use a small sample of all the data set to estimate the gradient, this is call stochastic gradient descent (SGD). Explain why we use SGD instead of gradient descent.

(c) Work out the following gradient: $\frac{\partial l}{\partial p_i}, \frac{\partial l}{\partial W_2}, \frac{\partial l}{\partial b_2}, \frac{\partial l}{\partial h_i}, \frac{\partial l}{\partial W_1}, \frac{\partial l}{\partial b_1}$. When deriving the gradient w.r.t. the parameters in lower layers, you can may assume the gradient in upper layers are available to you (i.e., you can use them in your equation). For example, when calculating $\frac{\partial l}{\partial W_1}$, you can assume $\frac{\partial l}{\partial p_i}, \frac{\partial l}{\partial W_2}, \frac{\partial l}{\partial b_2}, \frac{\partial l}{\partial h_i}$ are known.

3. Consider the following neural network for a 2-D input, $x_1 \in \mathbb{R}$ and $x_2 \in \mathbb{R}$ where:

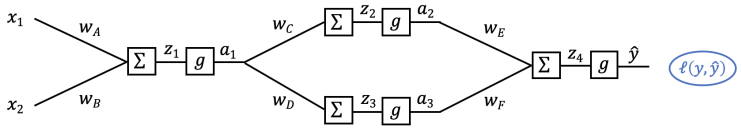


Figure 4: Neural Network

- All g functions are the same arbitrary non-linear activation function with no parameters
- $\ell(y, \hat{y})$ is an arbitrary loss function with no parameters, and:

$$z_1 = w_A x_1 + w_B x_2 \; a_1 = g(z_1)$$
$$z_2 = w_C a_1 \; a_2 = g(z_2)$$
$$z_3 = w_D a_1 \; a_3 = g(z_3)$$
$$z_4 = w_E a_2 + w_F a_3 \; \hat{y} = g(z_4)$$

Note: There are no bias terms in this network.

(a) What is the chain of partial derivatives needed to calculate the derivative $\frac{\partial \ell}{\partial w_E}$?

Your answer should be in the form: $\frac{\partial \ell}{\partial w_E} = \frac{\partial ?}{\partial ?}\frac{\partial ?}{\partial ?}\ldots$ Make sure each partial derivative $\frac{\partial ?}{\partial ?}$ in your answer cannot be decomposed further into simpler partial derivatives.

Do not evaluate the derivatives. Be sure to specify the correct subscripts in your answer.

$\frac{\partial \ell}{\partial w_E} =$

(b) The network diagram from above is repeated here for convenience: What is the

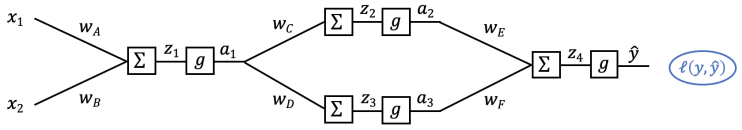


Figure 5: Neural Network

chain of partial deriviatives needed to calculate the derivative $\frac{\partial \ell}{\partial w_C}$?
Your answer should be in the form:

$$\frac{\partial \ell}{\partial w_C} = \frac{\partial ?}{\partial ?}\frac{\partial ?}{\partial ?}\ldots$$

Make sure each partial derivative $\frac{\partial ?}{\partial ?}$ in your answer cannot be decomposed further into simpler partial derivatives. **Do not evaluate the derivatives**. Be sure to specify the correct superscripts in your answer.

$\frac{\partial \ell}{\partial w_C} =$

(c) The gradient descent update step for weight w_c is:

$$w_c \leftarrow w_c - \alpha \frac{\partial Q}{\partial t} = \frac{\partial s}{\partial t}$$

where α (alpha) is the learning rate (step size).
Now, we want to change our neural network objective function to add an L2 regularization term on the weights. The new objective is:

$$\ell(y, \hat{y}) + \lambda \frac{1}{2} \|w\|_2^2$$

where λ (lambda) is the regularization hyperparamter and $\mathbf{w}$ is all of the weights in the neural network stacked into a single vector, $\mathbf{x} = [w_A, w_B, w_C, w_D, w_E, w_F]^T$.
Write the right-hand side of the new gradient descent update step for weight w_C given this new objective function. You may use $\frac{\partial \ell}{\partial w_C}$ in your answer.

Update: $w_C \leftarrow ...$

5 MLE/MAP

1. Please circle **True** or **False** for the following questions, providing brief explanations to support your answer.

   (i) **[2 pts]** Consider the linear regression model $y = w^T x + \epsilon$. Assuming $\epsilon \sim \mathcal{N}(0, \sigma^2)$ and maximizing the conditional log-likelihood is equivalent to minimizing the sum of squared errors $\|y - w^T x\|_2^2$.

   Circle one: True False
   One line justification (only if False):

   (ii) **[4 pts]** Consider n data points, each with one feature x_i and an output y_i. In linear regression, we assume $y_i \sim \mathcal{N}(wx_i, \sigma^2)$ and compute $\hat{w}$ through MLE.

   Suppose $y_i \sim \mathcal{N}(\log(wx_i), 1)$ instead. Then the maximum likelihood estimate $\hat{w}$ is the solution to the following equality:

   $$\sum_{i=1}^{n} x_i y_i = \sum_{i=1}^{n} x_i \log(wx_i)$$

   Circle one: **True** **False**
   Brief explanation:

2. **Select all that apply:** Which of the following are correct regarding Gradient Descent (GD). Assume data log-likelihood is $L(\theta|X)$, which is a function of the parameter θ, and the objective function is negative log-likelihood .

   - ☐ GD requires that $L(\theta|X)$ is concave with re sp ect to parameter θ in order to converge
   - ☐ GD requires that $L(\theta|X)$ is convex with re sp ect to parameter θ in order to converge
   - ☐ GD update rule is $\theta \leftarrow \theta - \alpha \nabla_\theta L(\theta|X)$
   - ☐ Given a fixed small learning rate (say $\alpha = 10^{-10}$), GD will always reach the optimum after infinite iterations (assume that the objective function satisfies the convergence condition).

3. Let $X_1, X_2, ..., X_N$ be i.i.d. data from a uniform distribution over a diamond-shaped area with edge length $\sqrt{2}\theta$ in $\mathbb{R}^2$, where $\theta \in \mathbb{R}^+$ (see Figure 6). Thus, $X_i \in \mathbb{R}^2$ and the distribution is

   $$p(x|\theta) = \begin{cases} \frac{1}{2\theta^2} & if \ \|x\| \leq \theta \\ 0 & otherwise \end{cases}$$

   where $\|x\| = |x_1| + |x_2|$ is $L1$ norm. Please find the maximum likelihood estimator of θ.

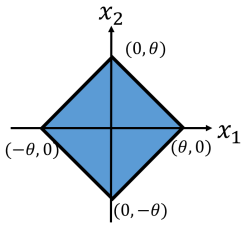


Figure 6: Area of $\|x\| \leq \theta$

4. **Short answer:** Suppose we want to model a 1-dimensional dataset of N real valued features $\left(x^{(i)}\right)$ and targets $\left(y^{(i)}\right)$ by:

$$y^{(i)} \sim \mathcal{N}\left(\exp(wx^{(i)}), 1\right)$$

Where w is our unknown (scalar) parameter and $\mathcal{N}$ is the normal distribution with probability density function:

$$f(a)_{\mathcal{N}(\mu,\sigma^2)} = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(a-\mu)^2}{2\sigma^2}\right)$$

Can the maximum conditional negative log likelihood estimator of w be solved analytically? If so, find the expression for w_{MLE}. If not, say so and write down the update rule for w in gradient descent.

5. Assume we have n iid random variables $x_i, i \in [1, n]$ such that each x_i belongs to a normal distribution with mean μ and variance σ^2.

$$p(x_1, x_2, ..., x_n|\mu, \sigma^2) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} exp\{\frac{-(x_i-\mu)^2}{2\sigma^2}\}$$

   a) Write the log likelihood function $l(x_1, x_2...x_n|\mu, \sigma^2)$

   b) Derive an expression for the Maximum Likelihood Estimate for the variance (σ^2)

6. Assume we have a random variable that is Bernoulli distributed $X_1, \ldots, X_n \sim Bernoulli(\theta)$. We are going to derive its MLE. Recall that in a Bernoulli $X = \{0, 1\}$ and the pdf of a Bernoulli is

$$p(X; \theta) = \theta^x (1-\theta)^{1-x}$$

   a) Derive the likelihood, $L(\theta; X_1, \ldots, X_n)$

   b) Derive the following formula for the log likelihood

$$l(\theta; X_1, \ldots, X_n) = (\sum_{i=1}^{n} X_i) \log(\theta) + (n - \sum_{i=1}^{n} X_i) \log(1-\theta)$$

   c) Derive the MLE, $\hat{\theta}$, and show that $\hat{\theta} = \frac{1}{n}(\sum_{i=1}^{n} X_i)$

7. Assume we have a random sample that is Bernoulli distributed $X_1, \ldots, X_n \sim \text{Exponential}(\theta)$. We are going to derive the MLE for θ. Recall that a exponential random variable X has p.d.f:

$$P(X; \theta) = \theta \exp(-\theta X).$$

   a) Derive the likelihood, $L(\theta; X_1, \ldots, X_n)$.

   b) Find θ that maximizes $L(\theta; X_1, \ldots, X_n)$.

8. For each question state **True** or **False** and give one line justifications.

   a) **T or F** The value of the Maximum Likelihood Estimate (MLE) is equal to the value of the Maximum A Posteriori (MAP) Estimate with a uniform prior.

   b) **T or F** The bias of the Maximum Likelihood Estimate (MLE) is typically less than or equal to the bias of the Maximum A Posteriori (MAP) Estimate.

   c) **T or F** The MAP estimate is always better than the MLE.

   d) **T or F** In the limit as n (the number of samples) increases, the MAP and MLE estimates become the same.

   e) **T or F** Naive Bayes can only be used with MAP estimates, and not MLE estimates.

6 Probability, Naive Bayes and MLE

6.1 Probability

1. For each question, circle the correct option.
    1. Which of the following expressions is equivalent to $p(A|B,C,D)$?
        (a) $\frac{p(A,B,C,D)}{p(C|B,D)p(B|D)p(D)}$
        (b) $\frac{p(A,B,C,D)}{p(B,C)p(D)}$
        (c) $\frac{p(A,B,C,D)}{p(B,C|D)p(B)p(C)}$
    2. Let μ be the mean of some probability distribution. $p(\mu)$ is always non-zero.
        (a) True
        (b) False
2. Assume we have a sample space Ω. Just state **T** or **F**, no justification needed.
    1. If events A, B, and C are disjoint then they are independent.
    2. $P(A|B) \propto \frac{P(A)P(B|A)}{P(A|B)}$.
    3. $P(A \cup B) \leq P(A)$.
    4. $P(A \cap B) \geq P(A)$.

6.2 Naive Bayes

1. Consider the following data. It has 4 features $\mathbf{X} = (x_1, x_2, x_3, x_4)$ and 3 labels $(+1, 0, -1)$. Assume that the probabilities $p(\mathbf{X}|y)$ and $p(y)$ are both Bernoulli distributions. Answer the questions that follow under the Naive Bayes assumption.

x_1	x_2	x_3	x_4	y
1	1	0	1	+1
0	1	1	0	+1
1	0	1	1	0
0	1	1	1	0
0	1	0	0	-1
1	0	0	1	-1
0	0	1	1	-1

    1. Compute the Maximum Likelihood Estimate for $p(x_i = 1|y), \forall i \in [1,4], \forall y \in \{+1, 0, -1\}$.
    2. Compute the Maximum Likelihood Estimate for the prior probabilities $p(y = +1), p(y = 0), p(y = -1)$

3. Use the values computed in the above two parts to classify the data point ($x_1 = 1, x_2 = 1, x_3 = 1, x_4 = 1$) as either belonging to class $+1, 0$ or -1

2. You are given a data set of 10,000 students with their sex, height, and hair color. You are trying to build a machine learning classifier to predict the sex of a student, so you randomly split the data into a training set and a testing set. Here are the specifications of the data set:

   - sex $\in$ {male,female}
   - height $\in$ [0,300] centimeters
   - hair $\in$ {brown, black, blond, red, green}
   - 3240 men in the data set
   - 6760 women in the data set

   Under the assumptions necessary for Naive Bayes (not the distributional assumptions you might naturally or intuitively make about the dataset) answer each question with **T** or **F** and a one sentence explanation of your answer:

   1. **T or F:** Height is a continuous valued variable. Therefore Naive Bayes is not appropriate since it cannot handle continuous valued variables.
   2. **T or F:** Since there is not a similar number of men and women in that dataset Naive Bayes will have high test error.
   3. **T or F:** $p(\texttt{height}|\texttt{sex}, \texttt{hair}) = p(\texttt{height}|\texttt{sex})$.
   4. **T or F:** $p(\texttt{height}, \texttt{hair}|\texttt{sex}) = p(\texttt{height}|\texttt{sex}) * p(\texttt{hair}|\texttt{sex})$.

6.3 Naive Bayes, Logistic Regression

1. Suppose you wish to learn $P(Y|X_1, X_2, X_3)$, where Y, X_1, X_2 and X_3 are all boolean-valued random variables. You consider both Naive Bayes and Logistic Regression as possible approaches.

   For each of the following, answer True or False, and give a *one sentence* justification for your answer.

   1. T or F: In this case, a good choice for Naive Bayes would be to implement a Gaussian Naive Bayes classifier.
   2. T or F: To learn $P(Y|X_1, X_2, X_3)$ using Naive Bayes, you must make conditional independence assumptions, including the assumption that Y is conditionally independent of X_1 given X_2.
   3. T or F: Logistic regression is certain to be the better choice in this case.

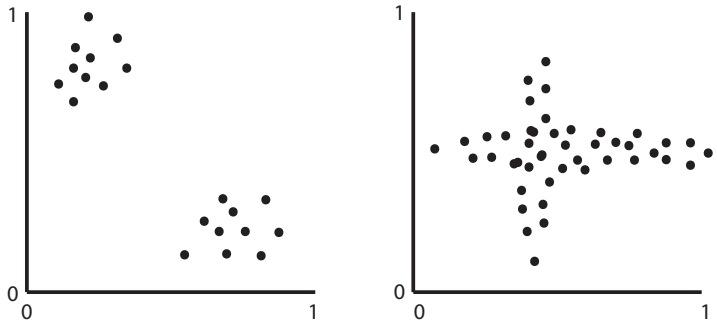
2. Parameter estimation

1. How many parameters must be estimated for your Gaussian Naive Bayes classifier, and what are they (i.e., please list them).
2. How many parameters must be estimated for your Logistic Regression classifier, and what are they (i.e., please list them).
3. T or F: We can train Naive Bayes using maximum likelihood estimates for each parameter, but not MAP estimates. Justify your answer *in one sentence*.
4. T or F: We can train Logistic Regression using maximum likelihood estimates for each parameter, but not MAP estimates. Justify your answer *in one sentence*.

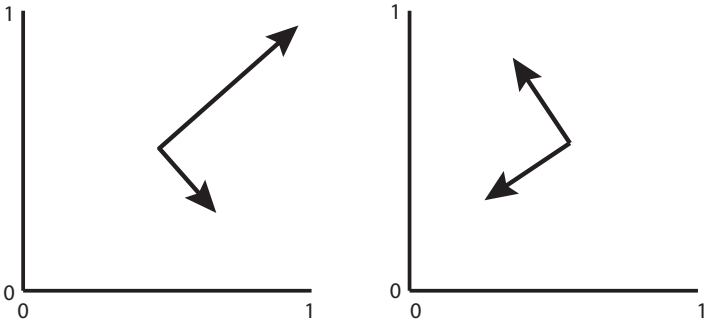
3. Mixing discrete and continuous variables. Suppose we add a numeric, real-valued variable X_4 to our problem. Note we now have a mix of some discrete-valued X_i and one continuous X_i.
   1. Explain in *two sentences* why we can no longer use Naive Bayes, or if we can, how we would modify our first solution.
   2. Explain in *two sentences* why we can no longer use Logistic Regression, or if we can, how we would modify our first solution.

7 Principal Component Analysis

1. (i) **[5 pts]** Consider the following two plots of data. Draw arrows from the mean of the data to denote the direction and relative magnitudes of the principal components.



(ii) **[5 pts]** Now consider the following two plots, where we have drawn only the principal components. Draw the data ellipse or place data points that could yield the given principal components for each plot. Note that for the right hand plot, the principal components are of equal magnitude.



2. Circle one answer and explain.

In the following two questions, assume that using PCA we factorize $X \in \mathbb{R}^{n \times m}$ as $Z^T U \approx X$, for $Z \in \mathbb{R}^{m \times n}$ and $U \in \mathbb{R}^{m \times m}$, where the rows of X contain the data points, the rows of U are the prototypes/principal components, and $Z^T U = \hat{X}$.

(i) **[2 pts]** Removing the last row of U and Z will still result in an approximation of X, but this will never be a better approximation than $\hat{X}$.

Circle one: True False

(ii) **[2 pts]** $\hat{X}\hat{X}^T = Z^T Z$.

Circle one: True False

(iii) **[2 pts]** The goal of PCA is to interpret the underlying structure of the data in terms of the principal components that are best at predicting the output variable.

Circle one: True False

(iv) **[2 pts]** The output of PCA is a new representation of the data that is always of lower dimensionality than the original feature representation.

Circle one: True False

8 K-Means

1. For **True or False** questions, circle your answer and justify it; for **QA** questions, write down your answer.

   (i) For a particular dataset and a particular k, k-means always produce the same result, if the initialized centers are the same. Assume there is no tie when assigning the clusters.

   ◯ True

   ◯ False

   Justify your answer:

   (ii) k-means can always converge to the global optimum.

   ◯ True

   ◯ False

   Justify your answer:

   (iii) k-means is not sensitive to outliers.

   ◯ True

   ◯ False

   Justify your answer:

   (iv) k in k-nearest neighbors and k-means has the same meaning.

   ◯ True

   ◯ False

   Justify your answer:

   (v) What's the biggest difference between k-nearest neighbors and k-means?

   Write your answer in one sentence:

   ---

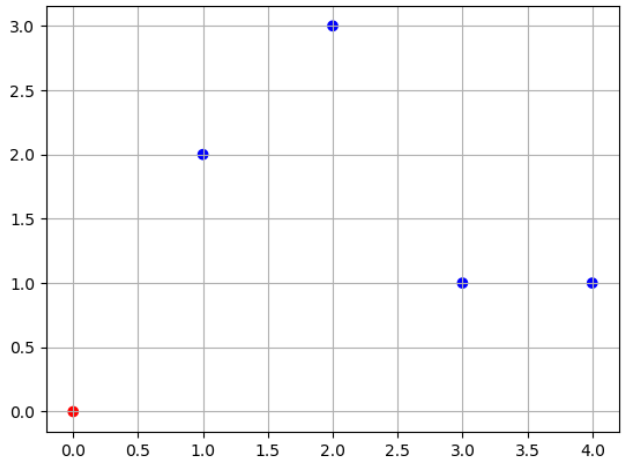
2. In k-means, random initialization could possibly lead to a local optimum with very bad performance. To alleviate this issue, instead of initializing all of the centers completely randomly, we decide to use a smarter initialization method. This leads us to k-means++.

   The only difference between k-means and k-means++ is the initialization strategy, and all of the other parts are the same. The basic idea of k-means++ is that instead of simply choosing the centers to be random points, we sample the initial centers iteratively, each time putting higher probability on points that are far from any existing center. Formally, the algorithm proceeds as follows.

   Given: Data set $x^{(i)}, i = 1, \ldots, N$
   Initialize:
   $\mu^{(1)} \sim \text{Uniform}(\{x^{(i)}\}_{i=1}^{N})$
   For $j = 2, \ldots, k$
   Computing probabilities of selecting each point
   $$p_i = \frac{\min_{j'<j} \|\mu^{(j')} - x^{(i)}\|_2^2}{\sum_{i'=1}^{N} \min_{j'<j} \|\mu^{(j')} - x^{(i')}\|_2^2}$$
   Select next center given the appropriate probabilities
   $\mu^{(j)} \sim \text{Categorical}(\{x^{(i)}\}_{i=1}^{N}, \mathbf{p}_{1:N})$

   Note: n is the number of data points, k is the number of clusters. For cluster 1's center, you just randomly choose one data point. For the following centers, every time you initialize a new center, you will first compute the distance between a data point and the center closest to this data point. After computing the distances for all data points, perform a normalization and you will get the probability. Use this probability to sample for a new center.

   Now assume we have 5 data points (n=5): (0, 0), (1, 2), (2, 3), (3, 1), (4, 1). The number of clusters is 3 (k=3). The center of cluster 1 is randomly choosen as (0, 0). These data points are shown in the figure below.



(i) **[5 pts]** What is the probability of every data point being chosen as the center for cluster 2? (The answer should contain 5 probabilities, each for every data point)

(ii) **[1 pts]** Which data point is mostly liken chosen as the center for cluster 2?

(iii) **[5 pts]** Assume the center for cluster 2 is chosen to be the most likely one as you computed in the previous question. Now what is the probability of every data point being chosen as the center for cluster 3? (The answer should contain 5 probabilities, each for every data point)

(iv) **[1 pts]** Which data point is mostly liken chosen as the center for cluster 3?

(v) **[3 pts]** Assume the center for cluster 3 is also chosen to be the most likely one as you computed in the previous question. Now we finish the initialization for all 3 centers. List the data points that are classified into cluster 1, 2, 3 respectively.

(vi) **[3 pts]** Based on the above clustering result, what's the new center for every cluster?

(vii) **[2 pts]** According to the result of (ii) and (iv), explain how does k-means++ alleviate the local optimum issue due to initialization?

9 Kernel Methods

1. **[3 pts.]** Applying the kernel trick enables features to be mapped into a higher dimensional space, at a cost of higher computational complexity to operate in the higher dimensional space.

   Circle one: True False

2. **[3 pts.]** Since the VC dimension for an SVM with a Radial Base Kernel is infinite, such an SVM must have a larger generalization error than an SVM without kernel which has a finite VC dimension.

   Circle one: True False

3. **[3 pts.]** Suppose $\phi(x)$ is an arbitrary feature mapping from input $x \in \mathcal{X}$ to $\phi(x) \in \mathbb{R}^N$ and let $K(x, z) = \phi(x) \cdot \phi(z)$. Then $K(x, z)$ will always be a valid kernel function.

   Circle one: True False

4. **[3 pts.]** Suppose $\phi(x)$ is the feature map induced by a polynomial kernel $K(x, z)$ of degree d, then $\phi(x)$ should be a d-dimensional vector.

   Circle one: True False

5. **[3 pts.]** The decision boundary that we get from a Gaussian Naive Bayes model with class-conditional variance is quadratic. Can we in principle reproduce this with an SVM and a polynomial kernel?

   Circle one: Yes No

6. Let's go kernelized!

   (a) **Perceptron review.** Assume we have a binary classification task with dataset $D = \{(x^{(i)}, y^{(i)})\}_{i=1}^{\infty}$ where $x^{(i)} \in \mathbb{R}^d$ and $y^{(i)} \in \{-1, 1\}$. Recall that the perceptron learns a linear classifier $y = sign(w^T x)$ by applying the following algorithm.

   Algorithm 1: Perceptron algorithm

   Initialize the weights $w = 0$;
   for $i = 1, 2, \cdots$ **do**
   | Predict $\hat{y}^{(i)} = sign(w^T x^{(i)})$;
   | **if** $\hat{y}^{(i)} \neq y^{(i)}$ **then**
   | | Update $w = w + y^{(i)} x^{(i)}$;
   end
   Final classifier: $h(x) = sign(w^T x)$

   Show that the final weight vector w is a linear combination of all the samples $x^{(i)}$ $(i = 1, 2, \cdots, T)$ it has been trained on, and hence for prediction we can write $w^T x$ in the form of $w^T x = \sum_{i=1}^{T} \alpha_i K(x^{(i)}, x)$ for some α_i where $K(x, z) = x^T z$.

(b) **Kernelized perceptron.** Now we are going to introduce a kernel function $K(x, z)$ to kernelize the perceptron algorithm. Based on your findings in the previous question, fill in the blanks below to complete the kernelized perceptron algorithm using the kernel $K(x, z)$. Assume the training loop stops after it has seen T training samples.

Algorithm 2: Kernelized Perceptron

Initialize _____;
for $i = 1, 2, \cdots$ **do**
  Predict $\hat{y}^{(i)} =$ _____;
  if $\hat{y}^{(i)} \neq y^{(i)}$ **then**
    _____;
end
Final classifier: $h(x) =$ _____.

(c) **Short answer.** Describe one advantage and one disadvantage of using kernelized perceptron compared to using vanilla perceptron.

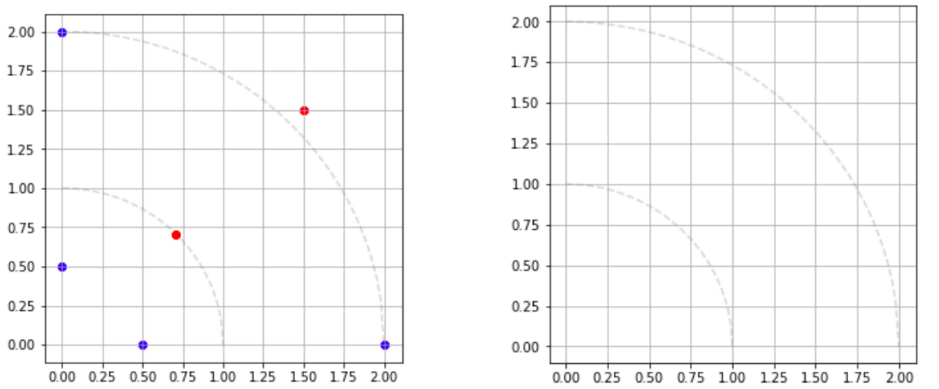
7. Suppose we have six training samples that lie in a two-dimensional space as is shown in Figure 7a. Four of them belong to the blue class: $(0, 0.5), (0, 2), (0.5, 0), (2, 0)$, and two of them belong to the red class: $(\sqrt{2}/2, \sqrt{2}/2), (1.5, 1.5)$. Unfortunately, this dataset is not linearly separable. You recall that kernel trick is one technique you can take advantage of to address this problem. The trick uses a kernel function $K(x, z)$ which implicitly defines a feature map $\phi(x)$ from the original space to the feature space. Consider the following normalized kernel:

$$K(x, z) = \frac{x^T z}{\|x\|_2 \|z\|_2}.$$

(a) What is the feature map $\phi(x)$ that corresponds to this kernel? Draw $\phi(x)$ for each training sample in Figure 7b.

(b) The samples should now be linearly separable in the feature space. The classifier in the feature space that gives the maximum margin can be represented as a line $w^T x + \alpha = 0$. Draw the decision boundary of this classifier in Figure 7b. What are the coefficients in the weight vector $w = (w_1, w_2)^T$? Hint: you don't need to compute them.

(c) Now we map the decision boundary obtained in (b) back to the original space. Write down the corresponding boundary in the original space in the format of an explicit equation. You can keep α in your equation. Try to plot its rough shape in Figure 7a.



(a) Data points in the original space (b) Data points in the feature space

Figure 7: Data points in two spaces

10 Recommender Systems

1. **[5pts]** Applied to the Netflix Prize problem, which of the following methods does NOT always require side information about the users and the movies?

   Select all that apply:

   ☐ Neighborhood methods

   ☐ Content filtering

   ☐ Latent factor methods

   ☐ Collaborative filtering

   ☐ None of the above

2. **[5pts] Select all that apply:**

   ☐ Using matrix factorization, we can embed both users and items in the same space

   ☐ Using matrix factorization, we can embed either solely users or solely items in the same space, as we cannot combine different types of data

   ☐ In a rating matrix of users by books that we are trying to fill up, the best-known solution is to fill the empty values with 0s and apply PCA, allowing the dimensionality reduction to make up for this lack of data

   ☐ Alternating minimization allows us to minimize over two variables

   ☐ Alternating minimization avoids the issue of getting stuck in local minima

   ☐ If the data is multidimensional, then overfitting is extremely rare

   ☐ Nearest neighbor methods in recommender systems are restricted to using euclidian distance for their distance metric

   ☐ None of the above

3. **[5pts]** Your friend Duncan wants to build a recommender system for his new website DuncTube, where users can like and dislike videos that are posted there. In order to build his system using collaborative filtering, he decides to use Non-Negative Matrix Factorization. What is an issue with Duncan's approach, and what could he change about the website *or* the algorithm in order to fix it?

4. [**3pts**] You and your friends want to build a movie recommendation system based on collaborative filtering. There are three websites (A, B and C) that you decide to extract users rating from. On website A, the rating scale is from 1 to 5. On website B, the rating scale is from 1 to 10. On website C, the rating scale is from 1 to 100. Assume you will have enough information to identify users and movies on one website with users and movies on another website. Would you be able to build a recommendation system? And briefly explain how would you do it?

5. [**3pts**] What is the difference between collaborative filtering and content filtering?

11 GMM and EM

1. Which of the quantities are updated in the E-step when EM algorithm is used to fit a Gaussian mixture model?

   Select all that apply.

   ☐ Mixture probabilities (sometimes denoted as π).

   ☐ Assignment probabilities (sometimes denoted as y or z).

   ☐ Cluster centers (sometimes denoted as μ).

   ☐ Cluster variances (sometimes denoted as σ or Σ).

   ☐ None of the above

2. Bernoulli Mixture Models

There are two coins $\mathcal{C}_1$ and $\mathcal{C}_2$ with unknown probabilities of heads, denoted by q_1 and q_2 respectively. $\mathcal{C}_1$ is chosen with probability τ and $\mathcal{C}_2$ is chosen with probability $1-\tau$. The chosen coin is flipped once and the result is recorded. The experiment is repeated five times and the final result is recorded as $X = \{H, H, T, H, T\}$. You wish to use the EM algorithm to estimate the parameters $\theta = [q_1, q_2, \tau]$. Suppose Z denotes the hidden variable where $z^{(i)} = 1$ if $\mathcal{C}_1$ was tossed for the i-th flip and 0 if $\mathcal{C}_2$ was tossed. You start off with an initial guess of $q_1^{(0)} = \frac{1}{4}, q_2^{(0)} = \frac{2}{3}$, and $\tau^{(0)} = \frac{1}{2}$.

In the following questions, please write your answer as an **exact irreducible fraction** $\frac{a}{b}$ where a, b are co-prime integers. For example, if your answer is 0.4125, please write it as $\frac{33}{80}$. The horizontal bar for the fraction $\frac{\square}{\square}$ is already present in the answer box.

(a) **[1 pt] E step:** Compute $p(z^{(i)} = 1 \mid x^{(i)} = H, \theta^{(0)})$

(b) **[1 pt] E step:** Compute $p(z^{(i)} = 1 \mid x^{(i)} = T, \theta^{(0)})$

(c) **[2 pt] M step:** Compute $\tau^{(1)}$ Hint: $\theta^{(t+1)} = \arg\max_\theta \mathbb{E}_{Z|X,\theta^{(t)}}[\log p(X, Z \mid \theta)]$

(d) **[2 pt] M step:** Compute $q_1^{(1)}$

(e) **[2 pt] M step:** Compute $q_2^{(1)}$

This is an optional workbox for the above questions. Feel free to leave this blank, we will only grade this to assign partial credit in the case your above answers are incorrect. If you choose to add work please label it clearly.