Homework 6: Learning Theory and Generative Models

10-301/10-601 Introduction to Machine Learning (Spring 2022)
https://www.cs.cmu.edu/~mgormley/courses/10601/
OUT: Friday, March 17th
DUE: Friday, March 24th
TAs: Abuzar, Aditya, Bhargav, Erin, Markov, Sami

Homework 6 covers topics on learning theory, MLE/MAP, Naive Bayes, CNNs, and RNNs. The homework includes multiple choice, True/False, and short answer questions. There will be no consistency points in general, so please make sure to double check your answers to all parts of the questions!

START HERE: Instructions

- **Collaboration Policy**: Please read the collaboration policy here: http://www.cs.cmu.edu/~mgormley/courses/10601/syllabus.html
- **Late Submission Policy: For this homework, you will only have 2 late days instead of the usual 3.** This allows us to provide feedback before the exam. See the late submission policy here: http://www.cs.cmu.edu/~mgormley/courses/10601/syllabus.html
- **Submitting your work:** You will use Gradescope to submit answers to all questions and code. Please follow instructions at the end of this PDF to correctly submit all your code to Gradescope.
  - **Written:** For written problems such as short answer, multiple choice, derivations, proofs, or plots, please use the provided template. Submissions can be handwritten onto the template, but should be labeled and clearly legible. If your writing is not legible, you will not be awarded marks. If your scanned submission misaligns the template, there will be a 5% penalty. Alternatively, submissions can be written in LaTeX. Each derivation/proof should be completed in the boxes provided. If you do not follow the template, your assignment may not be graded correctly by our AI assisted grader.

Instructions for Specific Problem Types

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

Select One: Who taught this course?

● Matt Gormley

○ 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?

● Henry Chai

○ 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?

10-601	10-~~6~~301

Written Questions (92 points)

1 LaTeX Bonus Point and Template Alignment (1 points)

1. (1 point) **Select one:** Did you use LaTeX for the entire written portion of this homework?

   ● Yes

   ○ No

2. (0 points) **Select one:** I have ensured that my final submission is aligned with the original template given to me in the handout file and that I haven't deleted or resized any items or made any other modifications which will result in a misaligned template. I understand that incorrectly responding yes to this question will result in a penalty equivalent to 2% of the points on this assignment.
   Note: Failing to answer this question will not exempt you from the 2% misalignment penalty.

   ● Yes

2 Convolutional Neural Network (14 points)

1. In this problem, consider a convolutional layer from a standard implementation of a CNN as described in lecture, without any bias term.

$$X = \begin{bmatrix} 1 & 0 & -2 & 3 & 4 & 1 \\ 2 & 9 & 5 & 6 & 0 & -1 \\ 0 & -3 & 1 & 3 & 4 & 4 \\ 6 & 5 & 2 & 0 & 6 & 8 \\ -5 & 4 & -3 & 1 & 3 & -2 \\ 4 & 1 & 2 & 8 & 9 & 7 \end{bmatrix} \qquad F = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \qquad Y = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix}$$

   (a) (1 point) Let an image X (6×6) be convolved with a filter F (3×3) using no padding and a stride of 1 to produce an output Y (4×4). What is value of j in the output Y?

   Your Answer

   8

   (b) (1 point) Suppose you instead had an input feature map (or image) of size 6×4 (height × width) and a filter of size 2×2, using no padding and a stride of 2, what would be the resulting output size? Write your answer in the format: height × width.

   Your Answer

   3×2

2. Parameter sharing is a very important concept for CNN because it drastically reduces the complexity of the learning problem and consequently that of the model required to tackle it. The following questions will deal with parameter sharing. Assume that there is no bias term in our convolutional layer.

   (a) (1 point) **Select all that apply:** Which of the following are parameters of a convolutional layer?

   □ Stride size

   □ Padding size

   □ Image size

   □ Filter size

   ■ Weights in the filter

   □ None of the above

   (b) (1 point) **Select all that apply:** Which of the following are hyperparameters of a convolutional layer?

   ■ Stride size

   ■ Padding size

   □ Image size

   ■ Filter size

   □ Weights in the filter

   □ None of the above

   (c) (1 point) Suppose for the convolutional layer, we are given grayscale images of size 22×22. Using one single 4×4 filter with a stride of 2 and no padding, what is the number of parameters you are learning in this layer?

   Your Answer

   16

   (d) (1 point) Now suppose we do not do parameter sharing. That is, each output pixel of this layer is computed by a separate 4×4 filter. Again we use a stride of 2 and no padding. What is the number of parameters you are learning in this layer?

   Your Answer

   1600

(e) (1 point) Now suppose you are given a 40×40 colored image, which consists of 3 channels, each representing the intensity of one primary color (so your input is a $40 \times 40 \times 3$ tensor). Once again, you attempt to produce an output map without parameter sharing, using a unique 4×4 filter per output pixel, with a stride of 2 and no padding. What is the number of parameters you are learning in this layer?

Your Answer

17328

(f) (1 point) In *one concise sentence*, describe a reason why parameter sharing is a good idea for a convolutional layer applied to image data, besides the reduction in number of learned parameters.

Your Answer

Parameter sharing in a convolutional layer is a good idea for image data because it enforces local connectivity, allowing the network to learn translation-invariant features that can be reused across different regions of the input image.

3. Neural the Narwhal was expecting to implement a CNN for Homework 5, but he is disappointed that he only got to write a simple fully-connected neural network.

(a) (2 points) Neural decides to implement a CNN himself and comes up with the following naive implementation:

```python
# image X has shape (H_in, W_in), and filter F has shape (K, K)
# the output Y has shape (H_out, W_out)
Y = np.zeros((H_out, W_out))
for r in range(H_out):
    for c in range(W_out):
        for i in range(K):
            for j in range(K):
                Y[r, c] += X[___blank___] * F[i, j]
```

What should be in the *blank* above so that the output `Y` is correct? Assume that `H_out` and `W_out` are pre-computed correctly.

Your Answer

```
(r+i, c+j)
```

(b) (2 points) Neural now wants to implement the backpropagation part of the network but is stuck. He decides to go to office hours to ask for help. One TA tells him that a CNN can actually be implemented using matrix multiplication. He receives the following 1D convolution example:

> Suppose you have an input vector $\mathbf{x} = [x_1, x_2, x_3, x_4, x_5]^T$ and a 1D convolution filter $\mathbf{w} = [w_1, w_2, w_3]^T$. Then if the output is $\mathbf{y} = [y_1, y_2, y_3]^T$, $y_1 = w_1 x_1 + w_2 x_2 + w_3 x_3$, $y_2 = \cdots$, $y_3 = \cdots$. If you look at this closely, this is equivalent to
>
> $$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \mathbf{A} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}$$
>
> where the matrix $\mathbf{A}$ is given as $\cdots$

What is matrix $\mathbf{A}$ for this $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{w}$? Write only the final answer. Your work will *not* be graded.

Your Answer

$$\begin{bmatrix} w_1 & w_2 & w_3 & 0 & 0 \\ 0 & w_1 & w_2 & w_3 & 0 \\ 0 & 0 & w_1 & w_2 & w_3 \end{bmatrix}$$

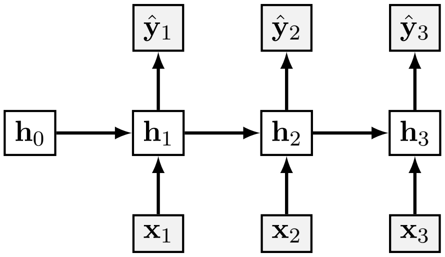
(c) (2 points) Neural wonders why the TA told him about matrix multiplication when he wanted to write the backpropagation part. Then he notices that the gradient is extremely simple with this version of CNN. Explain in *one concise sentence (or one short mathematical expression)* how you can compute $\frac{\partial \mathbf{y}}{\partial \mathbf{x}}$ once you obtain $\mathbf{A}$ for some *arbitrary* input $\mathbf{x}$, filter $\mathbf{w}$, and the corresponding 1D convolution output $\mathbf{y}$ (so $\mathbf{A}$ is obtained following the same procedure as in part (b), but $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{w}$ can be different from the example). Write only the final answer. Your work will *not* be graded.

Your Answer

$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \mathbf{A} \cdot \text{vec}(\mathbf{w})^T$ where $\text{vec}(\mathbf{w})$ is the vectorized form of $\mathbf{w}$ and $\mathbf{A}$ is the matrix obtained by performing the convolution operation with flipped $\mathbf{w}$.

3 Recurrent Neural Network (4 points)

1. Consider the following simple RNN architecture:



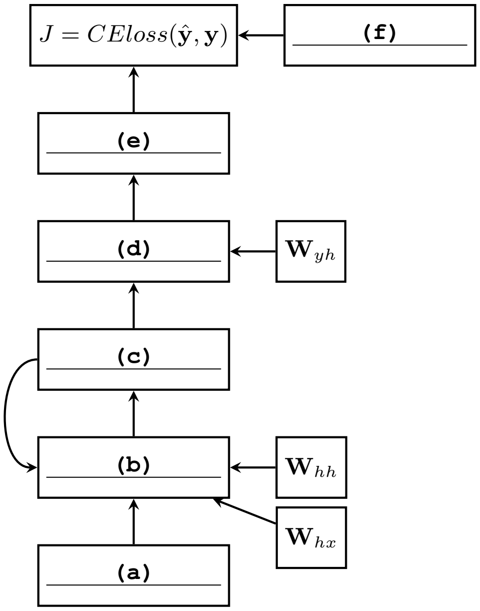
Where the layers and their corresponding weights are given below:

$$\begin{aligned} \mathbf{x}_t &\in \mathbb{R}^3 & \qquad \mathbf{W}_{hx} &\in \mathbb{R}^{4\times 3} \\ \mathbf{h}_t &\in \mathbb{R}^4 & \qquad \mathbf{W}_{yh} &\in \mathbb{R}^{2\times 4} \\ \mathbf{y}_t, \hat{\mathbf{y}}_t &\in \mathbb{R}^2 & \qquad \mathbf{W}_{hh} &\in \mathbb{R}^{4\times 4} \end{aligned}$$

$$\begin{aligned} J &= -\sum_{t=1}^{3}\sum_{i=1}^{2} y_{t,i}\log(\hat{y}_{t,i}) \\ \hat{\mathbf{y}}_t &= \sigma(\mathbf{o}_t) \\ \mathbf{o}_t &= \mathbf{W}_{yh}\mathbf{h}_t \\ \mathbf{h}_t &= \psi(\mathbf{z}_t) \\ \mathbf{z}_t &= \mathbf{W}_{hh}\mathbf{h}_{t-1} + \mathbf{W}_{hx}\mathbf{x}_t \end{aligned}$$

Where σ is the **softmax** activation and ψ is the **identity** activation (i.e. no activation). Note here that we assume that we have no intercept term. J here is computing the cross entropy loss.

(a) (3 points) You will now construct the computational graph for the given model. Use input $\mathbf{x}$, label $\mathbf{y}$, and the RNN equations presented above to complete the graph by filling in the solution boxes for the corresponding blanks.



(a)

Input $\mathbf{x}_1$

(b)

$\mathbf{h}_0 = \vec{0}$

(c)

$\mathbf{z}_1 = \mathbf{W}_{hx}\mathbf{x}_1$

(d)

$\mathbf{h}_1 = \psi(\mathbf{z}_1)$

(e)

$\mathbf{o}_1 = \mathbf{W}_{yh}\mathbf{h}_1$

(f)

$-\sum_{i=1}^{2} y_{3,i} \log(\hat{y}_{3,i})$

(b) (1 point) For this question, please write your answer in terms of W_{hh}, W_{yh}, y, $\hat{y}$, h, and any additional terms specified explicitly (note: this does not mean that every term listed shows up in the answer, but rather that you should simplify terms into these as much as possible when you can).

Suppose you have a variable $\mathbf{g}_{W_{hh},t}$ that stores the value of $\frac{\partial J_t}{\partial W_{hh}}$. What is $\frac{\partial J}{\partial \mathbf{W}_{hh}}$? Write your solution in terms of the $\mathbf{g}_{W_{hh},t}$ and the aforementioned variables in the first box, and show your work in the second.

$\frac{\partial J}{\partial W_{hh}}$

$\frac{\partial J}{\partial W_{hh}} = \sum_{t=1}^{3} \frac{\partial J_t}{\partial W_{hh}} = \mathbf{g}_{W_{hh},1} + \mathbf{g}_{W_{hh},2} + \mathbf{g}_{W_{hh},3}$

Work

4 Learning Theory (19 points)

1. Neural the Narwhal is given a classification task to solve, which he decides to use a decision tree learner with 2 binary features X_1 and X_2. On the other hand, you think that Neural should not have used a decision tree. Instead, you think it would be best to use logistic regression with 16 real-valued features in addition to a bias term. You want to use PAC learning to check whether you are correct. You first train your logistic regression model on N examples to obtain a training error $\hat{R}$.

   (a) (1 point) Which of the following case of PAC learning should you use for your logistic regression model?

   ○ Finite and realizable

   ○ Finite and agnostic

   ○ Infinite and realizable

   ● Infinite and agnostic

   (b) (2 points) What is the upper bound on the true error R in terms of $\hat{R}$, δ, and N? You may use big-$\mathcal{O}$ notation if necessary. Write only the final answer. Your work will *not* be graded.
   Note: Your answer may not contain any other symbols.

   Your Answer

$$R(h) \leq \hat{R}(h) + O\left(\sqrt{\frac{1}{N}\left(17 + \log\frac{1}{\delta}\right)}\right)$$

   (c) (3 points) **Select one:** You want to argue your method has a lower bound on the true error as compared to the Neural's true error bound. Assume that you have obtained enough data points to satisfy the PAC criterion with the same ϵ and δ as Neural. Which of the following is true?

   ○ Neural's model will always classify unseen data more accurately because it only needs 2 binary features and therefore is simpler.

   ○ You must first regularize your model by removing 14 features to make any comparison at all.

   ○ It is sufficient to show that the VC dimension of your classifier is higher than that of Neural's, therefore having a lower bound for the true error.

   ● It is necessary to show that the training error you achieve is lower than the training error Neural achieves.

2. In lecture, we saw that we can use our sample complexity bounds to derive bounds on the true error for a particular algorithm. Consider the sample complexity bound for the infinite, agnostic case:

$$N = O\left(\frac{1}{\epsilon^2}\left[\text{VC}(\mathcal{H}) + \log\frac{1}{\delta}\right]\right).$$

(a) (2 points) What is the big-$\mathcal{O}$ bound of ϵ in terms of N, δ, and VC$(\mathcal{H})$?
Note: $A = \mathcal{O}(B)$ (for some value B) $\Leftrightarrow$ there exists a constant $c \in \mathbb{R}$ such that $A \leq cB$.

Your Answer

$$\epsilon = \mathcal{O}\left(\frac{1}{\sqrt{\frac{N}{(\text{VC}(\mathcal{H})+\log\frac{1}{\delta})}}}\right)$$

(b) (2 points) Now, using the definition of ϵ (i.e. $|R(h) - \hat{R}(h)| \leq \epsilon$) and your answer to part a, prove that with probability at least $(1-\delta)$:

$$R(h) \leq \hat{R}(h) + O\left(\sqrt{\frac{1}{N}\left[\text{VC}(\mathcal{H}) + \log\frac{1}{\delta}\right]}\right).$$

Your Answer

By the definition of ϵ, we have:

$$|R(h) - \hat{R}(h)| \leq \epsilon. \tag{1}$$

From the previous question, we derived the big-$\mathcal{O}$ bound of ϵ:

$$\epsilon = \mathcal{O}\left(\frac{1}{\sqrt{\frac{N}{c(\mathrm{VC}(\mathcal{H})+\log\frac{1}{\delta})}}}\right). \tag{2}$$

Since $|R(h) - \hat{R}(h)| \leq \epsilon$ and we want to prove that with probability at least $(1 - \delta)$,

$$R(h) \leq \hat{R}(h) + O\left(\sqrt{\frac{1}{N}\left[\mathrm{VC}(\mathcal{H}) + \log\frac{1}{\delta}\right]}\right), \tag{3}$$

we can rewrite the inequality from Equation (1) as follows:

$$R(h) - \hat{R}(h) \leq \epsilon. \tag{4}$$

Now, substitute the bound of ϵ from Equation (2) into Equation (4):

$$R(h) - \hat{R}(h) \leq \mathcal{O}\left(\frac{1}{\sqrt{\frac{N}{\mathrm{VC}(\mathcal{H})+\log\frac{1}{\delta}}}}\right). \tag{5}$$

Next, we can simplify the expression inside the big-$\mathcal{O}$ notation:

$$R(h) - \hat{R}(h) \leq \mathcal{O}\left(\sqrt{\frac{1}{N}\left[\mathrm{VC}(\mathcal{H}) + \log\frac{1}{\delta}\right]}\right). \tag{6}$$

Finally, rearrange the inequality to get the desired result:

$$R(h) \leq \hat{R}(h) + \mathcal{O}\left(\sqrt{\frac{1}{N}\left[\mathrm{VC}(\mathcal{H}) + \log\frac{1}{\delta}\right]}\right. \tag{7}$$

3. (3 points) Consider the hypothesis space of functions that map M binary attributes to a binary label. A function f in this space can be characterized as $f : \{0,1\}^M \rightarrow \{0,1\}$. Neural the Narwhal says that regardless of the value of M, a function in this space can always shatter 2^M points. Is Neural wrong? If so, provide a counterexample. If Neural is right, briefly explain why in 1-2 *concise* sentences.

Your Answer

Neural the Narwhal is right. Given any set of 2^M points, we can construct a decision tree with M levels that correctly classifies all of them. Each level of the decision tree corresponds to a different attribute, and the decision at each level splits the set of points based on the value of that attribute. Since there are 2^M possible combinations of attribute values, the decision tree can classify all 2^M points. Since decision trees correspond to a hypothesis space of functions that map M binary attributes to a binary label, this shows that such a function can shatter 2^M points.

4. Consider an instance space $\mathcal{X}$ which is the set of real numbers.

   (a) (3 points) **Select one:** What is the VC dimension of hypothesis class H, where each hypothesis h in H is of the form "if $a < x < b$ or $c < x < d$ then $y = 1$; otherwise $y = 0$"? (i.e., H is an infinite hypothesis class where a, b, c, and d are arbitrary real numbers).

   ○ 2

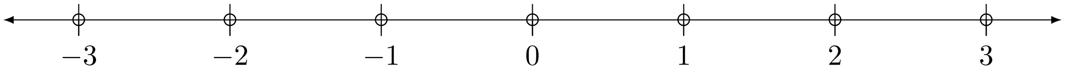
   ○ 3

   ● 4

   ○ 5

   ○ 6

   (b) (3 points) Given the set of points in $\mathcal{X}$ below, construct a labeling of some subset of the points to show that any dimension larger than the VC dimension of H by *exactly* 1 is incorrect (e.g. if the VC dimension of H is 3, only fill in the answers for 4 of the points). Fill in the boxes such that for each point in your example, the corresponding label is either 0 or 1. For points you are not using in your example, write N/A (do *not* leave the answer box blank).



Answer for -3	Answer for -2	Answer for -1
1	0	1

Answer for 0	Answer for 1	Answer for 2	Answer for 3
0	1	N/A	N/A

5 MLE/MAP (32 points)

1. (1 point) **True or False:** Suppose you place a Beta prior over the Bernoulli distribution, and attempt to learn the parameter θ of the Bernoulli distribution from data. Further suppose an adversary chooses "bad" but finite hyperparameters for your Beta prior in order to confuse your learning algorithm. As the number of training examples grows to infinity, the MAP estimate of θ can still converge to the MLE estimate of θ.

   ● True

   ○ False

2. (2 points) **Select one:** Let Γ be a random variable with the following probability density function (pdf):

$$f(\gamma) = \begin{cases} 2\gamma & \text{if } 0 \leq \gamma \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

   Suppose another random variable Y, which is conditioning on Γ, follows an exponential distribution with $\lambda = 3\gamma$. Recall that the exponential distribution with parameter λ has the following pdf:

$$f_{exp}(y) = \begin{cases} \lambda e^{-\lambda y} & \text{if } y \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

   What is the MAP estimate of γ given $Y = \frac{2}{3}$ is observed?

Your Answer
1

3. (4 points) Neural the Narwhal found a mystery coin and wants to know the probability of landing on heads by flipping this coin. He models the coin toss as sampling a value from Bernoulli(θ) where θ is the probability of heads. He flips the coin three times and the flips turned out to be heads, tails, and heads. An oracle tells him that $\theta \in \{0, 0.25, 0.5, 0.75, 1\}$, and *no other values of θ should be considered.*

   Find the MLE and MAP estimates of θ. Use the following prior distribution for the MAP estimate:

$$p(\theta) = \begin{cases} 0.9 & \text{if } \theta = 0 \\ 0.04 & \text{if } \theta = 0.25 \\ 0.03 & \text{if } \theta = 0.5 \\ 0.02 & \text{if } \theta = 0.75 \\ 0.01 & \text{if } \theta = 1 \end{cases}.$$

   Again, remember that $\theta \in \{0, 0.25, 0.5, 0.75, 1\}$, so the MLE and MAP should also be one of them.

MLE of θ	MAP of θ
0.5	0.5

4. In a previous homework assignment, you have derived the closed form solution for linear regression. Now, we are coming back to linear regression, viewing it as a statistical model, and deriving the MLE and MAP estimate of the parameters in the following questions.

As a reminder, in MLE, we have

$$\hat{\theta}_{MLE} = \underset{\theta}{\operatorname{argmax}}\, p(D|\theta)$$

For MAP, we have

$$\hat{\theta}_{MAP} = \underset{\theta}{\operatorname{argmax}}\, p(\theta|D)$$

Assume we have data $D = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^{N}$, where $\mathbf{x}^{(i)} = (x_1^{(i)}, \cdots, x_M^{(i)})$. So our data has N instances and each instance has M features. Each $y^{(i)}$ is generated given $\mathbf{x}^{(i)}$ with additive noise $\epsilon^{(i)} \sim \mathcal{N}(0, \sigma^2)$: that is, $y^{(i)} = \mathbf{w}^T\mathbf{x}^{(i)} + \epsilon^{(i)}$ where $\mathbf{w}$ is the parameter vector of linear regression.

(a) (2 points) **Select one:** Given this assumption, what is the distribution of $y^{(i)}$?

● $y^{(i)} \sim \mathcal{N}(\mathbf{w}^T\mathbf{x}^{(i)}, \sigma^2)$

○ $y^{(i)} \sim \mathcal{N}(0, \sigma^2)$

○ $y^{(i)} \sim \text{Uniform}(\mathbf{w}^T\mathbf{x}^{(i)} - \sigma, \mathbf{w}^T\mathbf{x}^{(i)} + \sigma)$

○ None of the above

(b) (2 points) **Select one:** The next step is to learn the MLE of the parameters of the linear regression model. Which expression below is the correct conditional log likelihood $\ell(\mathbf{w})$ with the given data?

● $\sum_{i=1}^{N}[-\log(\sqrt{2\pi\sigma^2}) - \frac{1}{2\sigma^2}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2]$

○ $\sum_{i=1}^{N}[\log(\sqrt{2\pi\sigma^2}) + \frac{1}{2\sigma^2}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2]$

○ $\sum_{i=1}^{N}[-\log(\sqrt{2\pi\sigma^2}) - \frac{1}{2\sigma^2}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})]$

○ $-\log(\sqrt{2\pi\sigma^2}) + \sum_{i=1}^{N}[-\frac{1}{2\sigma^2}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2]$

(c) (3 points) **Select all that apply:** Then, the MLE of the parameters is just $\operatorname{argmax}_{\mathbf{w}} \ell(\mathbf{w})$. Among the following expressions, select ALL that can yield the correct MLE.

□ $\operatorname{argmax}_{\mathbf{w}} \sum_{i=1}^{N}[-\log(\sqrt{2\pi\sigma^2}) - \frac{1}{2\sigma^2}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})]$

■ $\operatorname{argmax}_{\mathbf{w}} \sum_{i=1}^{N}[-\log(\sqrt{2\pi\sigma^2}) - \frac{1}{2\sigma^2}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2]$

■ $\operatorname{argmax}_{\mathbf{w}} \sum_{i=1}^{N}[-\frac{1}{2\sigma^2}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2]$

□ $\operatorname{argmax}_{\mathbf{w}} \sum_{i=1}^{N}[-\frac{1}{2}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})]$

■ $\operatorname{argmax}_{\mathbf{w}} \sum_{i=1}^{N}[-\frac{1}{2}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2]$

□ None of the above

5. Now we are moving on to learn the MAP estimate of the parameters of the linear regression model. Consider the same data D we used for the previous problem.

   (a) (3 points) **Select all that apply:** Which expression below is the correct optimization problem the MAP estimate is trying to solving? Recall that D refers to the data, and $\mathbf{w}$ to the regression parameters (weights).

   ■ $\mathbf{w}_{MAP} = \arg\max_{\mathbf{w}} p(D, \mathbf{w})$

   ■ $\mathbf{w}_{MAP} = \arg\max_{\mathbf{w}} \frac{p(D|\mathbf{w})p(\mathbf{w})}{p(D)}$

   □ $\mathbf{w}_{MAP} = \arg\max_{\mathbf{w}} \frac{p(D,\mathbf{w})}{p(\mathbf{w})}$

   ■ $\mathbf{w}_{MAP} = \arg\max_{\mathbf{w}} p(D|\mathbf{w})p(\mathbf{w})$

   ■ $\mathbf{w}_{MAP} = \arg\max_{\mathbf{w}} p(\mathbf{w}|D)$

   □ None of the above

   (b) (3 points) **Select one:** Suppose we are using a Gaussian prior distribution with mean 0 and variance $\frac{1}{\lambda}$ for each element w_m of the parameter vector $\mathbf{w}$, i.e. $w_m \sim \mathcal{N}\left(0, \frac{1}{\lambda}\right)$ $(1 \leq m \leq M)$. Assume that $w_1, \cdots, w_M$ are mutually independent of each other. Which expression below is the correct log joint-probability of the data and parameters $\log p(D, \mathbf{w})$? Please show your work below.

   ○ $\sum_{i=1}^{N}\left(-\log(\sqrt{2\pi\sigma^2}) - \frac{1}{2\sigma^2}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2\right) - \sum_{m=1}^{M}\log(\sqrt{2\pi\lambda}) - \lambda(w_m)^2$

   ○ $\sum_{i=1}^{N}\left(-\log(\sqrt{2\pi\sigma^2}) - \frac{1}{2\sigma^2}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})\right) + \sum_{m=1}^{M} -\log(\sqrt{2\pi\lambda}) - \lambda(w_m)^2$

   ○ $\sum_{i=1}^{N}\left(-\log(\sqrt{2\pi\sigma^2}) - \frac{1}{2\sigma^2}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})\right) - \sum_{m=1}^{M}\log(\sqrt{\frac{2\pi}{\lambda}}) - \frac{\lambda}{2}(w_m)^2$

   ● $\sum_{i=1}^{N}\left(-\log(\sqrt{2\pi\sigma^2}) - \frac{1}{2\sigma^2}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2\right) + \sum_{m=1}^{M} -\log(\sqrt{\frac{2\pi}{\lambda}}) - \frac{\lambda}{2}(w_m)^2$

Work

The log joint probability of the data and parameters is given by:

$$\begin{aligned}
\log p(D, \mathbf{w}) &= \log p(D|\mathbf{w}) + \log p(\mathbf{w}) \\
&= \log \left(\prod_{i=1}^{N} p(y^{(i)}|\mathbf{x}^{(i)}, \mathbf{w}) \right) + \log \left(\prod_{m=1}^{M} p(w_m) \right) \\
&= \sum_{i=1}^{N} \log p(y^{(i)}|\mathbf{x}^{(i)}, \mathbf{w}) + \sum_{m=1}^{M} \log p(w_m) \\
&= \sum_{i=1}^{N} \log \left(\frac{1}{\sqrt{2\pi\sigma^2}} \exp \left(-\frac{(y^{(i)} - \mathbf{w}^T \mathbf{x}^{(i)})^2}{2\sigma^2} \right) \right) + \sum_{m=1}^{M} \log \left(\frac{1}{\sqrt{2\pi/\lambda}} \exp \left(-\frac{\lambda (w_m)^2}{2} \right) \right) \\
&= \sum_{i=1}^{N} \left(-\log(\sqrt{2\pi\sigma^2}) - \frac{1}{2\sigma^2} (y^{(i)} - \mathbf{w}^T \mathbf{x}^{(i)})^2 \right) + \sum_{m=1}^{M} \left(-\log(\sqrt{2\pi/\lambda}) - \frac{\lambda}{2} (w_m)^2 \right) \\
&= \sum_{i=1}^{N} \left(-\log(\sqrt{2\pi\sigma^2}) - \frac{1}{2\sigma^2} (y^{(i)} - \mathbf{w}^T \mathbf{x}^{(i)})^2 \right) + \sum_{m=1}^{M} -\log(\sqrt{\frac{2\pi}{\lambda}}) - \frac{\lambda}{2} (w_m)^2
\end{aligned}$$

Therefore, the correct expression for the log joint-probability of the data and parameters is: $\sum_{i=1}^{N} \left(-\log(\sqrt{2\pi\sigma^2}) - \frac{1}{2\sigma^2} (y^{(i)} - \mathbf{w}^T \mathbf{x}^{(i)})^2 \right) + \sum_{m=1}^{M} -\log(\sqrt{\frac{2\pi}{\lambda}}) - \frac{\lambda}{2} (w_m)^2$

(c) (2 points) **Select one:** For the same linear regression model with a Gaussian prior on the parameters as in the previous question, maximizing the log posterior probability $\ell_{MAP}(\mathbf{w})$ gives you the MAP estimate of the parameters. Which of the following is an equivalent definition of $\max_{\mathbf{w}} \ell_{MAP}(\mathbf{w})$?

- ○ $\max_{\mathbf{w}} \sum_{i=1}^{N} \frac{1}{2\sigma^2}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2 + \frac{\lambda}{2}\|\mathbf{w}\|_2^2$
- ● $\min_{\mathbf{w}} \sum_{i=1}^{N} \frac{1}{2\sigma^2}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2 + \frac{\lambda}{2}\|\mathbf{w}\|_2^2$
- ○ $\max_{\mathbf{w}} \sum_{i=1}^{N} \frac{1}{2\sigma^2}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2 + \lambda\|\mathbf{w}\|_2^2$
- ○ $\min_{\mathbf{w}} -\sum_{i=1}^{N} \frac{1}{2\sigma^2}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2 - \frac{\lambda}{2}\|\mathbf{w}\|_2^2$

(d) (2 points) **Select one:** You found a MAP estimator that has a much higher test error than train error using some Gaussian prior. Which of the following could be a possible approach to fixing this?

- ○ Increase the variance of the prior used
- ● Decrease the variance of the prior used

6. (4 points) **Select one:** Suppose now the additive noise ϵ is different per datapoint. That is, each $y^{(i)}$ is generated given $\mathbf{x}^{(i)}$ with additive noise $\epsilon^{(i)} \sim \mathcal{N}(0, \sigma_i^2)$, i.e. $y^{(i)} = \mathbf{w}^T\mathbf{x}^{(i)} + \epsilon^{(i)}$. Unlike the standard regression model we have worked with until now, there is now an example specific variance σ_i^2. Maximizing the log-likelihood of this new model is equivalent to minimizing the *weighted* mean squared error with which of the following as the weights? Please show your work below.

- ○ $1/y^{(i)}$
- ● $1/\sigma_i^2$
- ○ $1/\|\mathbf{x}^{(i)}\|_2^2$

Work

The log-likelihood function of the new model is given by:

$$\begin{aligned}
\log p(D|\mathbf{w}) &= \sum_{i=1}^{N} \log p(y^{(i)}|\mathbf{x}^{(i)}, \mathbf{w}) \\
&= \sum_{i=1}^{N} \log \left(\frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2}{2\sigma_i^2} \right) \right) \\
&= \sum_{i=1}^{N} \left(-\log(\sqrt{2\pi\sigma_i^2}) - \frac{1}{2\sigma_i^2}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2 \right) \\
&= -\frac{1}{2}\sum_{i=1}^{N} \log(2\pi\sigma_i^2) - \frac{1}{2}\sum_{i=1}^{N} \frac{(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2}{\sigma_i^2}
\end{aligned}$$

Therefore, maximizing the log-likelihood of this new model is equivalent to minimizing the weighted mean squared error with weights $1/\sigma_i^2$.

7. (4 points) **Select one:** MAP estimation with what prior is equivalent to ℓ_1 regularization? Please show your work below.

   Note:

   - The pdf of a uniform distribution over $[a, b]$ is $f(x) = \frac{1}{b-a}$ if $x \in [a, b]$ and 0 otherwise.
   - The pdf of an exponential distribution with rate parameter a is $f(x) = a\exp(-ax)$ for $x > 0$.
   - The pdf of a Laplace distribution with location parameter a and scale parameter b is $f(x) = \frac{1}{2b}\exp\left(\frac{-|x-a|}{b}\right)$ for all $x \in \mathbb{R}$.

   ○ Uniform distribution over $[-1, 1]$

   ○ Uniform distribution over $\left[-\mathbf{w}^T\mathbf{x}^{(i)}, \mathbf{w}^T\mathbf{x}^{(i)}\right]$

   ○ Exponential distribution with rate parameter $a = \frac{1}{2}$

   ○ Exponential distribution with rate parameter $a = \mathbf{w}^T\mathbf{x}^{(i)}$

   ● Laplace distribution with location parameter $a = 0$

   ○ Laplace distribution with location parameter $a = \mathbf{w}^T\mathbf{x}^{(i)}$

Work

MAP estimation with a Laplace prior distribution with location parameter $a = 0$ is equivalent to ℓ_1 regularization.
The MAP estimation with a Laplace prior is given by:

$$\begin{aligned}
\hat{\mathbf{w}} &= \arg\max_{\mathbf{w}} p(\mathbf{w}|D) \\
&= \arg\max_{\mathbf{w}} p(D|\mathbf{w})p(\mathbf{w}) \\
&= \arg\max_{\mathbf{w}} \left(\prod_{i=1}^{N} p(y^{(i)}|\mathbf{x}^{(i)}, \mathbf{w})\right)\left(\prod_{m=1}^{M} p(w_m)\right) \\
&= \arg\max_{\mathbf{w}} \left(\prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2}{2\sigma^2}\right)\right)\left(\prod_{m=1}^{M} \frac{1}{2b}\exp\left(-\frac{|w_m|}{b}\right)\right) \\
&= \arg\min_{\mathbf{w}} \frac{1}{2\sigma^2}\sum_{i=1}^{N}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2 + \frac{1}{b}\sum_{m=1}^{M}|w_m| \\
&= \arg\min_{\mathbf{w}} \frac{1}{2\sigma^2}|\mathbf{y} - X\mathbf{w}|_2^2 + \lambda|\mathbf{w}|_1,
\end{aligned}$$

where $b = \lambda/\sigma^2$ is the scale parameter of the Laplace distribution, and λ is the regularization parameter. the Laplace prior with location parameter $a = 0$ is equivalent to ℓ_1 regularization

6 Naïve Bayes (22 points)

1. The following dataset describes several features of a narwhal and then whether or not it has an Instagram account.

Color	Size	Has Instagram?
Rainbow	Small	N
Cyan	Small	N
Cyan	Small	Y
Cyan	Medium	Y
Rainbow	Medium	N
Fuchsia	Medium	Y
Fuchsia	Large	Y
Cyan	Large	Y

Neural the Narwhal is cyan and medium-sized. We would like to determine whether he has an Instagram account, using the Naïve Bayes assumption to estimate the following probabilities.

(a) (2 points) What is the probability that a narwhal is cyan, medium-sized, and has an Instagram account? Round the answer to the fourth decimal place, e.g. 0.1234.

Your Answer

0.1172

(b) (2 points) What is the probability that a narwhal is cyan, medium-sized, and does *not* have an Instagram account? Round the answer to the fourth decimal place, e.g. 0.1234.

Your Answer

0.0703

(c) (1 point) **Select one:** Does Neural the Narwhal have an Instagram account?

● Yes

○ No

(d) (1 point) What is the generative story for this data? Name the distributions for the features and labels. Let Y represent whether or not a narwhal has an Instagram account and X_1 and X_2 represent Color and Size, respectively.

Y	$X_1\|Y$	$X_2\|Y$
$Y \sim Bernoulli(\theta)$ θ is the probability of a narwhal having an Instagram account	$P(X_1\|Y) \sim Categorical(\phi_{X_1\|Y})$ $\phi_{X_1\|Y}$ is the probability distribution of colors given the Instagram account status (Y)	$P(X_2\|Y) \sim Categorical(\phi_{X_2\|Y})$ $\phi_{X_2\|Y}$ is the probability distribution of sizes given the Instagram account status (Y)

(e) (1 point) How many parameters do we need to estimate?

Your Answer

9

(f) (1 point) Suppose we use Maximum Likelihood Estimation to train our model. What is our estimate of $\theta_{X_1=\text{ Rainbow},Y=N}$?

Your Answer

$\frac{1}{2}$

(g) (1 point) What is our MLE estimate of $\theta_{X_1=\text{ Fuchsia},Y=N}$?

Your Answer

$\frac{0}{2} = 0$

(h) (1 point) Give a test data point for which a Naïve Bayes model trained via MLE will never predict the correct label

Your Answer

Color = Rainbow, Size = Large, Has Instagram? = Y

2. (3 points) **Select all that apply:** Gaussian Naïve Bayes in general can learn non-linear decision boundaries. Consider the simple case where we have just one real-valued feature $X_1 \in \mathbb{R}$ from which we wish to infer the value of label $Y \in \{0, 1\}$. The corresponding generative story would be:

$$Y \sim \text{Bernoulli}(\phi)$$
$$X_1|Y \sim \text{Gaussian}\left(\mu_y, \sigma_y^2\right)$$

where the parameters are the Bernoulli parameter ϕ and the class-conditional Gaussian parameters μ_0, σ_0^2 and μ_1, σ_1^2 corresponding to $Y = 0$ and $Y = 1$, respectively.

A linear decision boundary in one dimension can be described by a rule of the form:

$$\text{if } X_1 > c \text{ then } Y = 1, \text{ else } Y = 0$$

where c is a real-valued threshold and $k \in \{0, 1\}$.

Is it possible in this simple one-dimensional case to construct a Gaussian Naïve Bayes classifier with a decision boundary that cannot be expressed by a rule in the above form? (Hint: Think about what a Gaussian distribution looks like for one random variable)

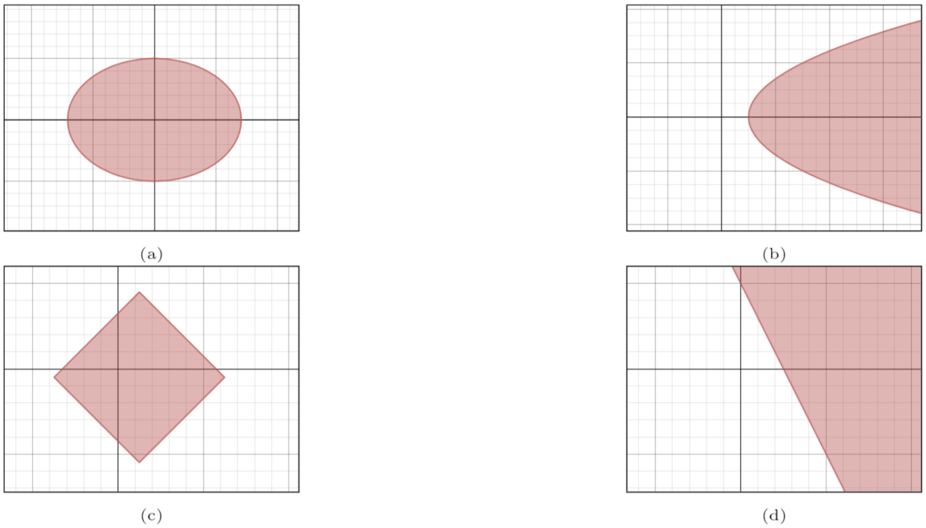
□ Yes, this can occur if the Gaussians are of equal means and unequal variances.

■ Yes, this can occur if the Gaussians are of unequal means and equal variances.

■ Yes, this can occur if the Gaussians are of unequal means and unequal variances.

□ None of the above

3. (2 points) **Select all that apply:** Select all possible decision boundaries that can be produced by a Gaussian Naïve Bayes classifier. The shaded region is assigned class 1 and the unshaded regions is assigned class 0. *(Hint: Recall the conclusion of the proof given in recitation.)*



(a)

(b)

(c)

(d)

Figure 1: Decision Boundaries

■ (a)

■ (b)

□ (c)

■ (d)

□ None of the above

4. Suppose we want to extend the Naïve Bayes model to time series data. Formally, a training data point consists of a binary label Y and D sequentially ordered observations of a binary outcome where X_1 occurs before X_2, which occurs before X_3 and so on all the way down to the final observation X_D; each feature X_d is binary.

You decide to modify the Naïve Bayes assumption such that a feature X_d is conditionally independent of all other features given the label Y and the previous feature X_{d-1}; the first feature X_1 is conditionally independent of all other features given just the label Y. The corresponding Generative story would be:

$$\begin{aligned} Y &\sim \text{Bernoulli}(\phi) \\ X_1|Y &\sim \text{Bernoulli}(\theta_1) \\ X_d|X_{d-1}, Y &\sim \text{Bernoulli}(\theta_{d,x,y}) \end{aligned}$$

where the parameters are the Bernoulli parameter ϕ and, the class-conditional Bernoulli parameter θ_1 , and the class-conditional Bernoulli parameters $\theta_{d,x,y}$ for $X_d|X_{d-1} = x, Y = y$.

(a) (2 points) Write down the expression for the joint distribution $P(X, Y)$ under your new Naïve Bayes model. You must use the modified Naïve Bayes assumption described above

Your Answer

Under the new Naïve Bayes model, the joint distribution $P(X, Y)$ can be expressed as:

$$\begin{aligned} P(X, Y) &= P(Y)P(X_1|Y) \prod_{d=2}^{D} P(X_d|X_{d-1}, Y) \\ &= \text{Bernoulli}(Y; \phi)\text{Bernoulli}(X_1; \theta_{1,Y}) \prod_{d=2}^{D} \text{Bernoulli}(X_d; \theta_{d,X_{d-1},Y}) \end{aligned}$$

where D is the total number of features, and the parameters ϕ, $\theta_{1,Y}$, and $\theta_{d,X_{d-1},Y}$ are the Bernoulli parameters for the respective distributions.

(b) (1 point) How many parameters do you need to learn in order to make predictions using this new Naïve Bayes model? Write your answer in terms of D.

Your Answer

4D - 2

(c) (2 points) Suppose we train this model via MLE. In at most 2 sentences, describe how we would estimate ϕ

Your Answer

Count the number of instances with Y=1 and divide it by the total number of instances in the dataset. This provides the proportion of instances with Y=1, which serves as our MLE estimate for ϕ.

(d) (2 points) In at most 2 sentences, describe how we would estimate $\theta_{d,x,y}$ for $2 \leq d \leq D$ using

MLE.

Your Answer

Count the number of instances with $X_d = 1$, $X_{d-1} = x$, and $Y = y$, and then divide this count by the total number of instances with $X_{d-1} = x$ and $Y = y$. This provides the conditional probability of $X_d = 1$ given $X_{d-1} = x$ and $Y = y$.

7 Collaboration Questions

After you have completed all other components of this assignment, report your answers to these questions regarding the collaboration policy. Details of the policy can be found here.

1. Did you receive any help whatsoever from anyone in solving this assignment? If so, include full details.
2. Did you give any help whatsoever to anyone in solving this assignment? If so, include full details.
3. Did you find or come across code that implements any part of this assignment? If so, include full details.

Your Answer

no, no, no