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: Thursday, October 27th
DUE: Friday, November 4th
TAs: Aditi, Chongling, Hang, Hayden, Sami

Homework 6 covers topics on learning theory, MLE/MAP, Naive Bayes, and CNNs. 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: https://www.cs.cmu.edu/~mgormley/courses/10601/syllabus.html
- **Late Submission Policy:** See the late submission policy here: https://www.cs.cmu.edu/~mgormley/courses/10601/syllabus.html
- **Submitting your work:** You will use Gradescope to submit your answers to all questions.
  - **Written:** For written problems such as short answer, multiple choice, derivations, proofs, or plots, please use the provided template. We strongly encourage you to typeset your submission using LaTeX. Submissions can also 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 submission is misaligned with the template, there will be a **5% penalty**. Each derivation/proof should be completed in the boxes provided. Do not move or resize any of the answer boxes. 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 (84 points)

1 LaTeX Bonus Point (1 points)

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

   ● Yes

   ○ No

2 Convolutional Neural Network (14 points)

1. In this problem, consider only the convolutional layer of a standard implementation of a CNN as described in lecture. We are given an image X and filter F below.

$X =$

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

$F =$

-1	-1	-1
-1	8	-1
-1	-1	-1

$Y =$

a	b	c	d
e	f	g	h
i	j	k	l
m	n	o	p

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

Your Answer

8

(b) (1 point) Suppose you had an input feature map of size (height × width) 6×4 and filter 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 x 2

2. Parameter sharing is a very important concept for CNN because it drastically reduces the complexity of the learning problem. For the following questions, 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 (so your input is a $40 \times 40 \times 3$ tensor), each representing the intensity of one primary color. Suppose you again do not do 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

By using parameter sharing, i.e. for a given activation map, the same kernel is used across the entire image, a feature detector that is useful in one part of the image is probably also useful in another part of the image, which means that the feature detectors can possibly capture local characters, e.g. edges, as well as global features, e.g. the hand of a person; In addition, we don't need to resize the parameters, if we change the image input size.

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:

```
# 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 $\mathbf{y} = [y_1, y_2, y_3]^T$, $y_1 = w_1x_1 + w_2x_2 + w_3x_3$, $y_2 = \cdots$, $y_3 = \cdots$. 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

$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \mathbf{A}^T$

3 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.

   Your Answer

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

   (c) (3 points) **Select one:** You want to argue your method has a lower bound on the true error. 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:

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

(a) (2 points) Rewrite the big-$\mathcal{O}$ bound in terms of N and δ using the definition of big-$\mathcal{O}$ notation (i.e. if $N = O(M)$ (for some value M), then there exists a constant $c \in \mathbb{R}$ such that $N \leq cM$).

Your Answer

From $N = O\left(\frac{1}{\epsilon^2}\left[\mathrm{VC}(\mathcal{H}) + \log\frac{1}{\delta}\right]\right)$, we know that there exists a constant $c \in \mathbb{R}$ such that $N \leq c \cdot \frac{1}{\epsilon^2}\left[\mathrm{VC}(\mathcal{H}) + \log\frac{1}{\delta}\right]$).
$\Rightarrow \epsilon^2 \leq c \cdot \frac{1}{N}\left[\mathrm{VC}(\mathcal{H}) + \log\frac{1}{\delta}\right] \Rightarrow \epsilon \leq \sqrt{c \cdot \frac{1}{N}\left[\mathrm{VC}(\mathcal{H}) + \log\frac{1}{\delta}\right]}$, since $\epsilon \geq 0$.
$\Rightarrow \epsilon \leq c_1\sqrt{\cdot\frac{1}{N}\left[\mathrm{VC}(\mathcal{H}) + \log\frac{1}{\delta}\right]} \Rightarrow \epsilon \leq \mathcal{O}(\sqrt{\frac{1}{N}\left[\mathrm{VC}(\mathcal{H}) + \log\frac{1}{\delta}\right]})$

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

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

Your Answer

From the definition we know that infinite, agnostic case, for any hypothesis set hypothesis set H and distribution p^*, if the sample complexity bound satisfies $N = O\left(\frac{1}{\epsilon^2}\left[\mathrm{VC}(\mathcal{H}) + \log\frac{1}{\delta}\right]\right)$, then with probability at leat $1 - \delta$, all $h \in H$ have $|R(h) - \hat{R}(h)| \leq \epsilon$.
From question (a), rewrite the big-$\mathcal{O}$ bound, we can get $\epsilon \leq \mathcal{O}(\sqrt{\frac{1}{N}\left[\mathrm{VC}(\mathcal{H}) + \log\frac{1}{\delta}\right]})$
$\Rightarrow R(h) - \hat{R}(h) \leq |R(h) - \hat{R}(h)| \leq \epsilon \leq \mathcal{O}(\sqrt{\frac{1}{N}\left[\mathrm{VC}(\mathcal{H}) + \log\frac{1}{\delta}\right]})$
$\Rightarrow R(h) \leq \hat{R}(h) + \mathcal{O}(\sqrt{\frac{1}{N}\left[\mathrm{VC}(\mathcal{H}) + \log\frac{1}{\delta}\right]})$

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 is right, since we can use the M binary features to represent the 2^M cases, i.e. each of 2^M points with label either 0 or 1. Then regardless of the value of M, the defined hypothesis space can always 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

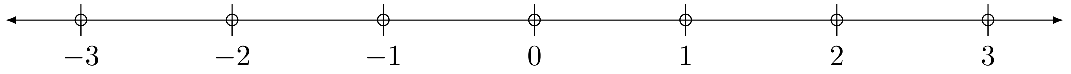
○ 3

● 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
N/A	1	0

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

4 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
0.75	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

$$\log p(D, \mathbf{w}) = \log p(D|\mathbf{w}) + \log p(\mathbf{w}) \quad (1)$$

$$= \log p(D|\mathbf{w}) + \log p(w_1) + \log p(w_2) ... + \log p(w_M) \quad (2)$$

since $w_1, ... w_M$ are mutually independent of each other.

$$= \log p(D|\mathbf{w}) + \sum_{m=1}^{M} \log p(w_m) \quad (3)$$

$$= \log p(D|\mathbf{w}) + \sum_{m=1}^{M} \log\left(\frac{1}{\sqrt{2\pi\frac{1}{\lambda}}} e^{-\frac{1}{2}\frac{w_m^2}{\frac{1}{\lambda}}}\right) \quad (4)$$

$$= \sum_{i=1}^{N} \left(-\log(\sqrt{2\pi\sigma^2}) - \frac{1}{2\sigma^2}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2\right) + \sum_{m=1}^{M} -\log\sqrt{\frac{2\pi}{\lambda}} - \frac{\lambda}{2}w_m^2 \quad (5)$$

(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

$$\operatorname*{argmax}_{\mathbf{w}} \sum_{i=1}^{N} \log p(y^{(i)}|\mathbf{w}) = \operatorname*{argmax}_{\mathbf{w}} \sum_{i=1}^{N} \log\Big(\frac{1}{\sigma_i\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)}}{\sigma_i})^2}\Big) \tag{6}$$

, since $\mathrm{y}^{(i)} \sim \mathcal{N}(\mathbf{w}^T\mathbf{x}^{(i)}, \sigma_i^2)$.

$$= \operatorname*{argmax}_{\mathbf{w}} \sum_{i=1}^{N} [-\log(\sqrt{2\pi\sigma_i^2}) - \frac{1}{2\sigma_i^2}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2] \tag{7}$$

$$= \operatorname*{argmax}_{\mathbf{w}} \sum_{i=1}^{N} [-\frac{1}{\sigma_i^2}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2] \tag{8}$$

$$= \operatorname*{argmin}_{\mathbf{w}} \sum_{i=1}^{N} [\frac{1}{\sigma_i^2}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2] \tag{9}$$

$$\tag{10}$$

Therefore, the weights are $\frac{1}{\sigma_i^2}$.

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

Let prior as a laplace distribution with location parameter $a = 0 : p(w_m) = \frac{1}{2b}\exp\left(\frac{-|w_m|}{b}\right)$.

$$\underset{\mathbf{w}}{\text{argmax}} \log p(D, \mathbf{w}) = \log p(D|\mathbf{w}) + \log p(\mathbf{w}) \tag{11}$$

$$= \underset{\mathbf{w}}{\text{argmax}} \log p(D|\mathbf{w}) + \log p(w_1) + \log p(w_2)... + \log p(w_M) \tag{12}$$

since $w_1, ...w_M$ are mutually independent of each other.

$$= \underset{\mathbf{w}}{\text{argmax}} \log p(D|\mathbf{w}) + \sum_{m=1}^{M} \log p(w_m) \tag{13}$$

$$= \underset{\mathbf{w}}{\text{argmax}} \log p(D|\mathbf{w}) + \sum_{m=1}^{M} \log\left(\frac{1}{2b}\exp\left(\frac{-|w_m|}{b}\right)\right) \tag{14}$$

$$= \underset{\mathbf{w}}{\text{argmax}} \sum_{i=1}^{N}\left(-\frac{1}{2\sigma^2}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2\right) - \log(\frac{M}{2b})\frac{1}{b}\sum_{m=1}^{M}|w_m| \tag{15}$$

$$= \underset{\mathbf{w}}{\text{argmin}} \sum_{i=1}^{N}\frac{1}{2\sigma^2}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2 + \log(\frac{M}{2b})\frac{1}{b}\sum_{m=1}^{M}|w_m| \tag{16}$$

$$= \underset{\mathbf{w}}{\text{argmin}} \sum_{i=1}^{N}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2 + \frac{2\sigma^2}{b}\log(\frac{M}{2b})\sum_{m=1}^{M}|w_m| \tag{17}$$

$$= \underset{\mathbf{w}}{\text{argmin}} \sum_{i=1}^{N}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2 + \lambda\sum_{m=1}^{M}|w_m|, let\ \lambda = \frac{2\sigma^2}{b}\log(\frac{M}{2b}) \tag{18}$$

(19)

5 Naïve Bayes (18 points)

1. (3 points) Suppose that 0.3% of all people have cancer. If someone is tested for cancer, the outcome of the test can either be positive (cancer) or negative (no cancer). The test is not perfect—among people who have cancer, the test comes back positive 97% of the time. Among people who don't have cancer, the test comes back positive 4% of the time. For this question, you should assume that the test results are independent of each other, given the true state (cancer or no cancer). What is the probability of a test subject having cancer, given that the subject's test result is positive? Round the answer to the fourth decimal place, e.g. 0.1234.

   Your Answer

   0.0680

2. 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

   0.1500

   (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

   0.0417

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

   ● Yes

   ○ No

3. (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:

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

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:

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

where c is a real-valued threshold and $k \in \{0, 1\}$. Note that $X_1 > c$ here denotes a generic linear inequality. That is, we use $X_1 > c$ to mean any of $X_1 > c$, $X_1 \geq c$, $X_1 < c$, and $X_1 \leq c$.

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?

■ 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

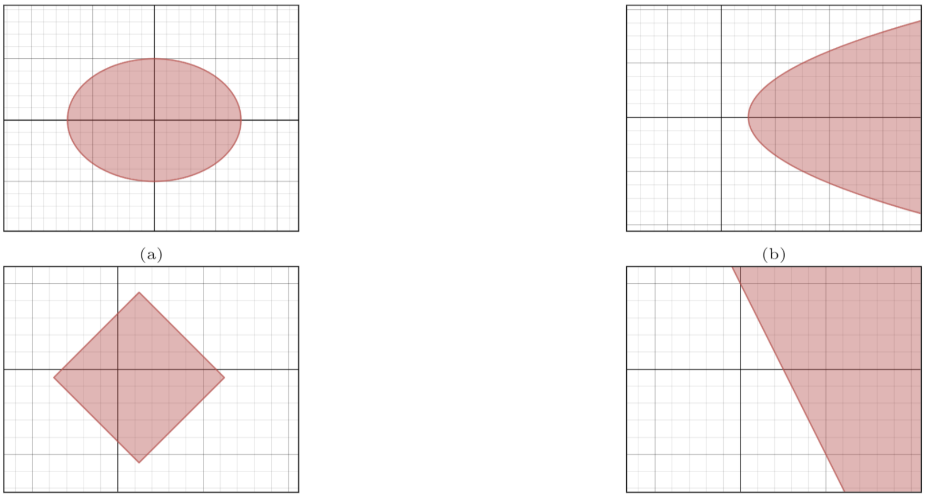
4. Now we will expand this to 2D and derive the decision boundary of a 2D Gaussian Naïve Bayes.

   (a) (3 points) Show that the decision boundary of a 2D Gaussian Naïve Bayes classifier is quadratic. That is, show that $p(y = 1 \mid x_1, x_2) = p(y = 0 \mid x_1, x_2)$ can be written as a polynomial function of x_1 and x_2 where the degree of each variable is at most 2. You may fold *unimportant* constants into terms such as C, C', C'', C''' so long as *you are clearly showing each step*.

Your Answer

$p(x_d|y = k) = \frac{1}{\sigma_{d,k}\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu_{d,k}}{\sigma_{d,k}})^2}$
Since $p(y = 1|x_1, x_2) = \frac{p(y=1,x_1,x_2)}{p(x_1,x_2)}$, and $p(y = 0|x_1, x_2) = \frac{p(y=0,x_1,x_2)}{p(x_1,x_2)}$,
then to solve $p(y = 1|x_1, x_2) = p(y = 0|x_1, x_2)$ is equivalent to solve $p(y = 1, x_1, x_2) = p(y = 0, x_1, x_2)$.
$p(y = 1, x_1, x_2) = p(x_1, x_2|y = 1)p(y = 1) = p(x_1|y = 1)p(x_2|y = 1)p(y = 1)$ (by Naïve Bayes Assumption)
dsw_1
$= e^{-\frac{1}{2}(\frac{x_1-\mu_{1,1}}{\sigma_{1,1}})^2 - \frac{1}{2}(\frac{x_2-\mu_{2,1}}{\sigma_{2,1}})^2 (C_1)'}$
Similarly, we can get $p(y = 0, x_1, x_2) = e^{-\frac{1}{2}(\frac{x_1-\mu_{1,0}}{\sigma_{1,0}})^2 - \frac{1}{2}(\frac{x_2-\mu_{2,0}}{\sigma_{2,0}})^2 (C_2)'}$.
Then $p(y = 1, x_1, x_2) = p(y = 0, x_1, x_2) \Rightarrow -\frac{1}{2}(\frac{x_1-\mu_{1,1}}{\sigma_{1,1}})^2 - \frac{1}{2}(\frac{x_2-\mu_{2,1}}{\sigma_{2,1}})^2 (C_1)' = -\frac{1}{2}(\frac{x_1-\mu_{1,0}}{\sigma_{1,0}})^2 - \frac{1}{2}(\frac{x_2-\mu_{2,0}}{\sigma_{2,0}})^2 (C_2)'$, which is a polynomial function of x_1 and x_2 where the degree of each variable is at most 2.

(b) (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.



(a)

(b)

(c)

(d)

Figure 1: Decision Boundaries

■ (a)

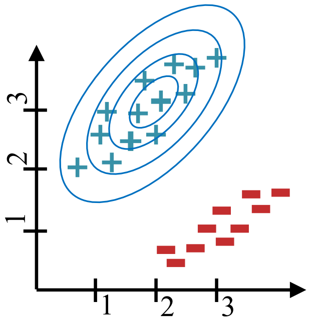
■ (b)

□ (c)

■ (d)

□ None of the above

5. (2 points) **Select all that apply:** Neural the Narwhal used a 2D Gaussian Naïve Bayes model for a binary classification task, and obtained the following contour plot for the density $f(x_1, x_2 \mid y = +)$. You instantly notice that he made a mistake. Why is his result theoretically impossible to be obtained from a Gaussian Naïve Bayes model?



□ The mean of the distribution should be the mean of all points, not just positive points (+)

□ A Bernoulli Naïve Bayes model should be used instead for discrete labels (+ and −)

■ The covariance matrix of the distribution is not a diagonal matrix

□ None of the above

6 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

1. No 2. No 3. No