HOMEWORK 6
LEARNING THEORY AND GENERATIVE MODELS *

10-301 / 10-601 INTRODUCTION TO MACHINE LEARNING (FALL 2021)
http://mlcourse.org

OUT: Oct. 21, 2021
DUE: Oct. 28, 2021
TAs: Catherine, Zachary, Ari, Siyuan, Anoushka

Homework 6 covers topics on learning theory, MLE/MAP, Naive Bayes, and revisits neural networks, logistic regression and regularization. The homework includes multiple choice, True/False, and short answer 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:** 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.

*Compiled on Thursday 28th October, 2021 at 23:23

Instructions for Specific Problem Types

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

Select One: Who taught this course?

● Matt Gormley / Henry Chai

○ 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?

● Matt Gormley / 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

☐ None of the above

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.

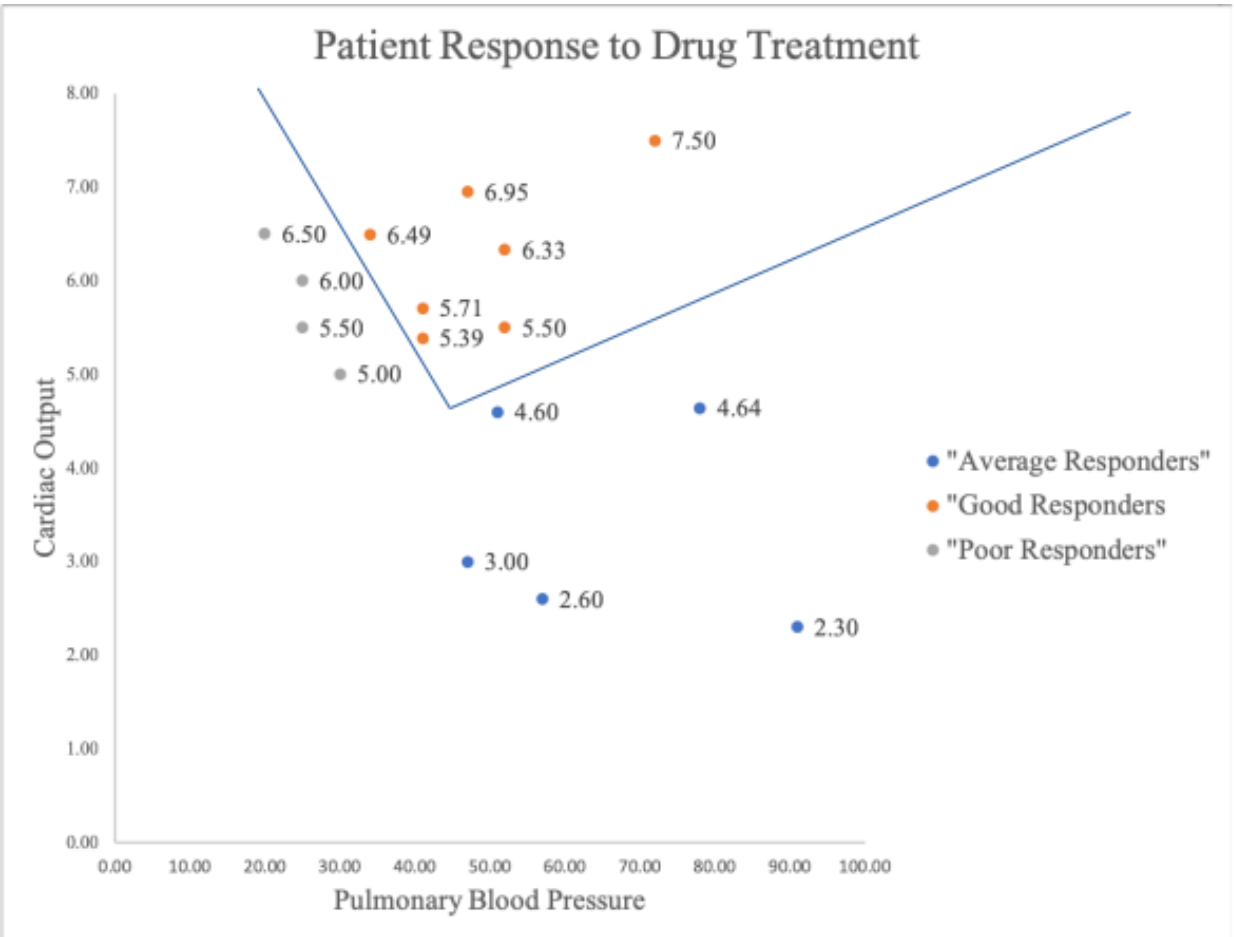
Fill in the blank: What is the course number?

10-601

10-~~7~~601

1 Neural Networks, Logistic Regression, Regularization Revisited

Imagine you work for a pharmaceutical company that is trying to predict whether certain patients will respond well to a new drug. Specifically, these patients have high blood pressure in their lungs, a condition known as pulmonary hypertension. Doctors recommend that the best predictors of a treatment effect is the patient's heart function (measured by cardiac output) and the blood pressure in their lungs (pulmonary blood pressure). You plot the data and visualize the following:



1. (3 points) **[SOLO]** Draw on the graph above the decision boundaries of a trained neural network that minimizes the training error when classifying Good Responders vs all others (Poor or Average). Assume the neural network has two hidden units and one hidden layer. What is the smallest training error you can achieve?

   Fill in the blank (write answer as a fraction):

   0

2. (2 points) **[SOLO]** Using your decision boundaries above, assuming a logistic activation function, which point has the highest probability of being a Good Responder? Provide the point number as shown on the graph (points are labeled by their y value).

   Fill in the blank:

7.50

3. (2 points) **[SOLO]** True or False: Increasing the number of hidden units of a neural network will always guarantee a lower training error.

   ◯ True

   ● False

4. (2 points) **[SOLO]** Convolutional neural networks often consist of convolutional layers, max-pooling layers, and fully-connected layers. Select all the layer types below that have parameters (i.e. weights) which can be learned by gradient descent / backpropagation.

   Select all that apply:

   ■ convolutional layer

   □ max-pooling layer

   ■ fully-connected layer

5. (2 points) **[SOLO]** Regularization.

   Which of the following are true about regularization? **Select all that apply:**

   ■ One of the goals of regularization is combating overfitting.

   □ A model with regularization fits the training data better than a model without regularization

   ■ The L-0 norm (number of non-zero parameters) is rarely used in practice in part because it is non-differentiable.

   ■ One way to understand regularization is that it attempts to follow Occam's razor and make the learning algorithm prefer "simpler" solutions.

6. (6 points) **[SOLO]** Regularization in Linear Regression.

   When performing linear regression, which of the following options will **not** increase mean-squared training error:

   Select all that apply:

   ■ Adding higher-order functions of the features as separate features

   □ Increasing the regularization weight

   □ For the same weight on the regularizer, using an L1 regularizer instead of an L2

   □ For the same weight on the regularizer, using an L1 regularizer instead of an L0

   □ None of the above

7. (3 points) **[GROUP]** Recall the convolution operation in CNN from lecture. A kernel, or a convolutional filter, is a small matrix that can be used for blurring, sharpening, embossing, edge detection, etc. Given that convolving the below black-and-white picture (labeled as Original) with kernel X gives output image (a), choose the most probable output image for kernel Y.

Original

(a)

(b)

(c)

-1	-2	-1
0	0	0
1	2	1

X

-1	0	1
-2	0	2
-1	0	1

Y

Select one:

- ◯ (a)
- ● (b)
- ◯ (c)
- ◯ None of the above

2 Learning Theory

1. **[SOLO]** Alex is given a classification task to solve.

   (a) (3 points) **[SOLO]** He has no idea where to start, so he decided to try out a decision tree learner with 2 binary features X_1 and X_2. He recently learned about PAC learning, and would like to know what is the minimum number (N) of data points that would suffice for the PAC criterion with $\epsilon = 0.1$ and $\delta = 0.01$.

   Notice that a valid decision tree may or may not be full, meaning it doesn't have to split on all features.

   Fill in the blank:

   404

   (b) (2 points) **[SOLO]** Sally thinks Alex shouldn't have used a decision tree with 2 binary features. Instead, she thinks it would be best to use logistic regression with 16 real-valued features in addition to a bias term. Sally overherd Alex talking about this cool concept called PAC learning and she too would like to use it to analyze her method. She first trains her logistic regression model on N examples to obtain a training error $\hat{R}$. What is the the upper bound on the true error R in terms of $\hat{R}$, δ, and N. You may use big-$\mathcal{O}$ notation.

   Fill in the blank:

   Upper bound on R,

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

   (c) (3 points) **[SOLO]** Sally wants to argue her method has lower bound on the true error. Assuming Sally has obtained enough data points to satisfy PAC criterion with $\epsilon = 0.1$ and $\delta = 0.01$. Which of the following is true?

   Select one:

   - ◯ Sally is wrong. Alex's method will always classify unseen data more accurately since it is simpler as it only needs 2 binary features.
   - ◯ She must first regularize her model by removing 14 features to make any comparison at all.
   - ◯ It is sufficient to show that the VC Dimension of her classifier is higher than Alex's, therefore having lower bound for the true error.
   - ● It is necessary to show that the training error she achieves is lower than the training error Alex achieves.

2. (4 points) **[SOLO]** In lecture, we saw that we can use our sample complexity bounds to derive bounds on the true error for a particular algorithm. Use the sample complexity bound for the infinite, agnostic case:

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

to prove that with probability at least $(1-\delta)$:

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

Hint: Start by applying the definition of big-O notation (i.e. if $N = O(M)$ (for some value M), then there exists a $c \in \mathbb{R}$ such that $N \leq cM$).

Your Answer

Given,

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

$$\therefore N \leq c\left(\frac{1}{\epsilon^2}\left[\text{VC}(\mathcal{H}) + \log(\frac{1}{\delta})\right]\right)$$

where $c \in \mathbb{R}$ is some constant
Rearranging leads to

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

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

From PAC criterion, with at least a probability of $1-\delta$,

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

$$R(h) \leq \hat{R}(h) + \epsilon \quad (2)$$

Substituting bound from (1) in (2)

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

3. (3 points) **[GROUP]** Consider an arbitrary decision tree learner applied to data where each example is described by M binary attributes, with no overlapping points (i.e. no points have identical attributes but different labels). Your friend Paul says that no matter the value of M, a decision tree can always shatter 2^M points. Is Paul wrong? If so, provide a counterexample. If Paul is right, briefly explain why in 1-2 *concise* sentences.

Your Answer

I believe Paul is right. A decision tree can split on one of the attributes and then split on some other attribute next along each tree path. Thus, the decision tree can end up forming a balanced binary tree with a capacity of having 2^M specific leaves, if required. This allows it to satisfy any labelling for 2^M data points where each point has M attributes.

4. **[GROUP]** Consider instance space $\mathcal{X}$ which is the set of real numbers.

   (a) (3 points) **[GROUP]** 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).

   Select one:

   ○ 2

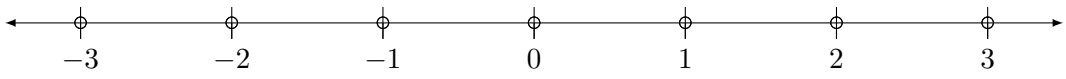
   ○ 3

   ● 4

   ○ 5

   ○ 6

   (b) (3 points) **[GROUP]** 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 your choice of VC dimension in part (a) 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 1 or 0 (for points you are not using in your example, leave the boxes blank).



-3: [ ]

-2: [1]

-1: [0]

0: [1]

1: [0]

2: [1]

3: [ ]

3 MLE/MAP

1. (3 points) **[SOLO] True or False:** Suppose you place a Beta prior over the Bernoulli distribution, and attempt to learn the parameter 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.

   Select One:

   - ● True
   - ○ False

2. (3 points) **[SOLO]** Let Θ be a random variable with the following probability density function (pdf):

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

   Suppose another random variable Y, which is conditioning on Θ, follows an exponential distribution with $\lambda = 3\theta$. 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 Θ given $Y = \frac{2}{3}$ is observed?

   Select one:

   - ○ 0
   - ○ 1/3
   - ● 1
   - ○ 2

3. (3 points) **[SOLO]** In HW3, 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

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

   For MAP, we have

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

   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 attributes/features. Each $y^{(i)}$ is generated given $\mathbf{x}^{(i)}$ with additive noise $\epsilon^{(i)} \sim$

$N(0, \sigma^2)$, that is $y^{(i)} = \mathbf{w}^T\mathbf{x}^{(i)} + \epsilon^{(i)}$ where $\mathbf{w}$ is the parameter vector of linear regression. Given this assumption, what is the distribution of y?

Select one:

- ● $y^{(i)} \sim N(\mathbf{w}^T\mathbf{x}^{(i)}, \sigma^2)$
- ○ $y^{(i)} \sim N(0, \sigma^2)$
- ○ $y^{(i)} \sim \text{Uniform}(\mathbf{w}^T\mathbf{x}^{(i)} - \sigma, \mathbf{w}^T\mathbf{x}^{(i)} + \sigma)$
- ○ None of the above

4. (3 points) **[SOLO]** 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?

Select one:

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

5. (3 points) **[SOLO]** Then, the MLE of the parameters is just $\text{argmax}_{\mathbf{w}}\, \ell(\mathbf{w})$. Among the following expressions, select ALL that can yield the correct MLE.

Select all that apply:

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

6. (3 points) **[SOLO]** According to the above derivations, is the MLE for the conditional log likelihood equivalent to minimizing mean squared errors (MSE) for the linear regression model when making predictions? Why or why not?

Select one:

- ○ Yes, because the derivative of the negative conditional log-likelihood has the same form as the derivative of the MSE loss.
- ● Yes, because the parameters that maximize the conditional log-likelihood also minimize the MSE loss.
- ○ No, because one is doing maximization and the other is doing minimization.
- ○ No, because the MSE has an additional error term $\epsilon^{(i)}$ in the expression whereas the quantity to be minimized in MLE does not.

◯ No, because the conditional log-likelihood has additional constant terms that do not appear in the MSE loss.

7. (3 points) **[SOLO]** Now we are moving on to learn the MAP estimate of the parameters of the linear regression model. 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)).

   Select all that apply:

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

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

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

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

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

8. (3 points) **[SOLO]** 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 $\mathbf{w}(1 \leq m \leq M)$, i.e. $w_m \sim N(0, \frac{1}{\lambda})$. 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.

   Select one:

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

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

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

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

9. (3 points) **[SOLO]** Maximizing the log posterior probability $\ell_{MAP}(\mathbf{w})$ gives you the MAP estimate of the parameters. Which one is correct to estimate the parameters using $\max_{\mathbf{w}} \ell_{MAP}(\mathbf{w})$ based on your derived log posterior probability in the previous question? With the result, please specify how the MAP estimate with Gaussian prior related to the linear regression model.

   Select one:

   ◯ $\max_{\mathbf{w}} \frac{1}{2}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2 + \frac{\lambda}{2}\|\mathbf{w}\|_2$

   ● $\min_{\mathbf{w}} \frac{1}{2}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2 + \frac{\lambda}{2}\|\mathbf{w}\|_2^2$

   ◯ $\max_{\mathbf{w}} \frac{1}{2}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2 + \lambda\|\mathbf{w}\|_2$

   ◯ $\min_{\mathbf{w}} -\frac{1}{2}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2 - \frac{\lambda}{2}\|\mathbf{w}\|_2^2$

Your answer

The MAP estimate with Gaussian prior assumed that the weights, w, came from a Gaussian distribution with mean 0 and variance $\frac{1}{\lambda}$. By doing so, it assumes that the values of the weights would mostly be spread around 0 with a variance $\frac{1}{\lambda}$. So, while the conditional log probability of D given w gives the loss term equivalence of MSE that linear regression uses, thr prior information of w being spread around 0 enforces a L2 regularization term as part of the objective function. This term ensures that the max_w would have a low value spread around 0 with a variance of $\frac{1}{\lambda}$.

10. (2 points) **[GROUP]** A MAP estimator with a Gaussian prior $\mathcal{N}(0, \sigma^2)$ you trained gives significantly higher test error than train error. What could be a possible approach to fixing this?

Select one:

- ◯ Increase variance σ^2
- ● Decrease variance σ^2
- ◯ Try MLE estimator instead
- ◯ None of the above

11. (4 points) **[GROUP]** MAP estimation with what prior is equivalent to L1 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 $x \in \mathbb{R}$.

Select one:

- ◯ Uniform distribution over $[-\mathbf{w}^T\mathbf{x}^{(i)}, \mathbf{w}^T\mathbf{x}^{(i)}]$
- ◯ Exponential distribution with rate parameter $a = \frac{1}{2}$
- ◯ Exponential distribution with rate parameter $a = \mathbf{w}^T\mathbf{x}^{(i)}$
- ● Laplace prior with location parameter $a = 0$
- ◯ Laplace prior with location parameter $a = \mathbf{w}^T\mathbf{x}^{(i)}$
- ◯ Uniform distribution over [-1, 1]

Your answer

Consider a Laplace prior with location parameter $a = 0$. So, the prior in the MAP log estimate is given by

$$l_w = argmax_w \log \left[\left(\prod_{j=1}^{M} \frac{1}{2b} e^{-\frac{|w_j - 0|}{b}} \right) \right]$$

$$l_{MAP} = argmax_w \sum_{j=1}^{M} \left[\log(\frac{1}{2b}) - \frac{1}{b}|w_j| \right]$$

$$l_{MAP} = argmax_w M \log(\frac{1}{2b}) - \frac{1}{b} \sum_{j=1}^{M} |w_j|$$

For the purpose of finding parameters maximizing the above terms, we can remove all the constant (with respect to w) terms. Therefore, ignore terms - $M \log(\frac{1}{2b})$.

$$\therefore l_{MAP} = argmax_w - \frac{1}{b} \sum_{j=1}^{M} |w_j|$$

Therefore, maximizing above is the same as minimizing the negative of above term.

$\therefore l_{MAP} = argmin_w \lambda \sum_{j=1}^{M} |w_j|$

From the above, it can be seen that the regularization/penalizing term generated by a Laplace prior with $a = 0$ is a L1 regularization with the regularization parameter $\lambda = \frac{1}{b}$

12. (2 points) **[GROUP]** When we estimate linear regression, we naturally choose the objective function of mean square error: $MSE(\theta; \mathcal{D}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \theta^T \mathbf{x}_i)^2$. We could instead minimize the weighted mean squared error: $WMSE(\theta; \mathcal{D}, \mathbf{a}) = \frac{1}{n} \sum_{i=1}^{n} a_i (y_i - \theta^T \mathbf{x}_i)^2$ where $a_1, ...a_n$ are fixed weights of the training examples that are given alongside the training data. This includes ordinary least squares as the special case where all the weights $a_i = 1$. Please suggest one reason that we would prefer minimizing weighted mean square error instead of just mean square error.

Your answer

Consider a set of training data of size 1000 corresponding to the task of Anomaly Detection. Since the positive cases (anomalies) are rare, the overall number of dataset with $y^{(i)} = 1$ is very less as compared to that equalling 0. So, say 5 training data points correspond to positive cases.
Now with an objective function such as MSE, the linear regression estimate could end up marking every training point as 0 and would still end up with a very low train error (0.5%). However, as per the task misclassifying the anomalies could lead to some really serious issues. So, as a need to ensure that the model is punished more for misclassifying an anomaly, the corresponding objective term needs to be weighted more.

Generalizing, a WMSE allows us to provide the right importance for each data point, thereby providing a prior knowledge regarding the data point distribution rather than treating them all equally.

13. (4 points) **[GROUP]** Suppose we define a new regression model. Each $y^{(i)}$ is generated given $\mathbf{x}^{(i)}$ with additive noise $\epsilon^{(i)} \sim N(0, \sigma_i^2)$, that is $y^{(i)} = \mathbf{w}^T\mathbf{x}^{(i)} + \epsilon^{(i)}$. Unlike the standard regression model we've worked with until now, there is now an example specific variance σ_i^2.

    Show that maximizing the log-likelihood of this new model is equivalent to minimizing a weighted mean squared error in the box below. Then select the correct value of a_i in the weighted mean squared error.

    Select one:

    - ○ $\frac{1}{y_i}$
    - ● $\frac{1}{\sigma_i^2}$
    - ○ $\frac{1}{x_i^2}$
    - ○ $\frac{1}{\theta_i}$

    Justification

Justification

$$l_{MLE} = argmax_w \log(\frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(y^{(i)} - w^T x^{(i)})^2)}{2\sigma_i^2}})$$

$$l_{MLE} = argmax_w - \log(\sqrt{2\pi\sigma_i^2}) - \frac{1}{2\sigma_i^2}(y^{(i)} - w^T x^{(i)})^2$$

Maximizing above log likelihood is the same as minimizing the negative log likelihood.

$$l_{MLE} = argmin_w \log(\sqrt{2\pi\sigma_i^2}) + \frac{1}{2\sigma_i^2}(y^{(i)} - w^T x^{(i)})^2$$

Removing terms which are constant with respect to the parameter (w).

$$\therefore l_{MLE} = argmin_w \frac{1}{2\sigma_i^2}(y^{(i)} - w^T x^{(i)})^2$$

The $\frac{1}{2}$ is common across all i examples. So, the weight for the ith data point is $\frac{1}{\sigma_i^2}$

4 Naive Bayes

1. (3 points) **[SOLO]** I give you the following fact: for events A and B, $P(A \mid B) = 2/3$ and $P(A \mid \neg B) = 1/3$, where $\neg B$ denotes the complement of B. Do you have enough information to calculate $P(B \mid A)$? If not, choose "not enough information", if so, compute the value of $P(B \mid A)$.

   Select one:

   ◯ 1/2

   ◯ 2/3

   ◯ 1/3

   ● Not enough information

2. (3 points) **[SOLO]** Instead if I give you for events A and B, $P(A \mid B) = 2/3$, $P(A \mid \neg B) = 1/3$ and $P(B) = 1/3$ and $P(A) = 4/9$, where $\neg B$ denotes the complement of B. Are the information consistent to calculate $P(B \mid A)$? If not, choose "conflicting information", if so, compute the value of $P(B \mid A)$.

   Select one:

   ● 1/2

   ◯ 2/3

   ◯ 1/3

   ◯ Conflicting information

3. (4 points) **[SOLO]** Suppose that 0.3% people have cancer. Someone decided to take a medical test 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?

   If your answer is in decimals, answer with precision 4, e.g. (6.051, 0.1230, 1.234e+7)

   Fill in the blank:

   0.0680

4. (3 points) **[SOLO]** Which of the following machine learning algorithms are probabilistic generative models?

   Select one:

   ◯ Decision Tree

   ◯ K-nearest neighbors

   ◯ Perceptron

   ● Naive Bayes

   ◯ Logistic Regression

◯ Feed-forward neural network

5. (2 points) **[SOLO]** Select all possible decision boundary 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.

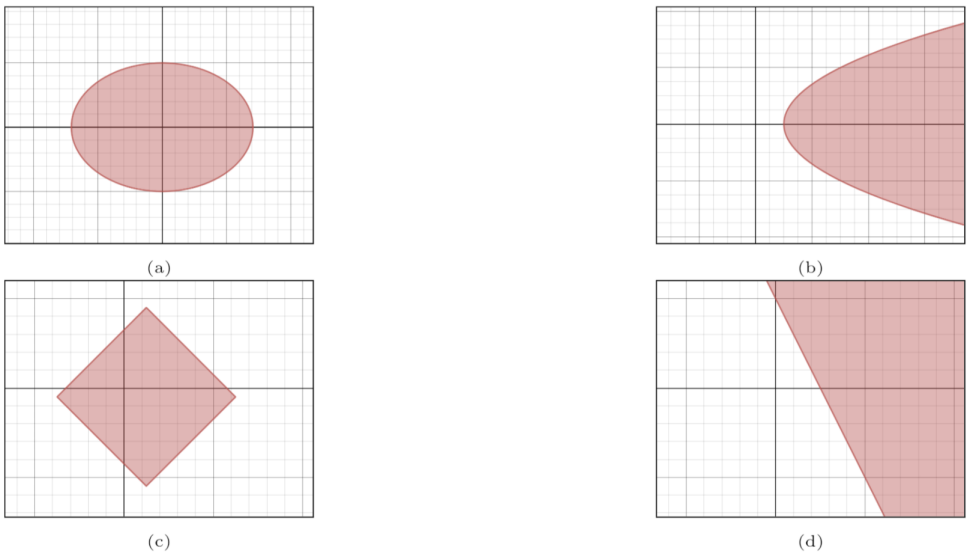


Figure 1: Decision Boundaries

Select all that apply:

■ (a)

■ (b)

□ (c)

■ (d)

□ None of the above

6. (4 points) **[GROUP]** Logistic Regression and Naive Bayes.

When Y is Boolean and $\mathbf{X} = \langle X_1 ... X_n \rangle$ is a vector of continuous variables, then the assumptions of the Gaussian Naive Bayes classifier imply that $P(Y \mid \mathbf{X})$ is given by the logistic function with appropriate parameters w_i for all i and b. In particular:

$$P(Y = 1 \mid \mathbf{X}) = \frac{1}{1 + \exp(b + \sum_{i=1}^{n} w_i X_i)}$$

and

$$P(Y = 0 \mid \mathbf{X}) = \frac{\exp(b + \sum_{i=1}^{n} w_i X_i)}{1 + \exp(b + \sum_{i=1}^{n} w_i X_i)}$$

Consider instead the case where Y is Boolean and $\mathbf{X} = \langle X_1 ... X_n \rangle$ is a vector of Boolean variables.

Since the X_i are Boolean variables, you need only one parameter to define $P(X_i \mid Y = y_k)$. Define $\phi_{i1} \equiv P(X_i = 1 \mid Y = 1)$, in which case $P(X_i = 0 \mid Y = 1) = (1 - \phi_{i1})$. Similarly, use ϕ_{i0} to denote $P(X_i = 1|Y = 0)$.

1. Show that

$$P(Y = 1|X) = \frac{1}{1 + \exp(\ln(\frac{1-\pi}{\pi}) + \sum_i [X_i \ln(\phi_{i0}) + \ln(1 - \phi_{i0}) - X_i \ln(1 - \phi_{i0}) - X_i \ln(\phi_{i1}) - \ln(1 - \phi_{i1}) + X_i \ln(1 - \phi_{i1}))]}$$

can be written in the form of a Gaussian Naive Bayes classifier by finding expressions for $P(Y = 1|X)$ and $P(Y = 0|X)$ in terms of b and w_i. Explicitly define b and w_i.

$$P(Y=1|X) = \frac{P(Y=1)\prod_{i=1}^{n} P(X_i|Y=1)}{\prod_{i=1}^{n} P(X_i)}$$

$$P(Y=1|X) = \frac{P(Y=1)\prod_{i=1}^{n} P(X_i|Y=1)}{P(Y=1)\prod_{i=1}^{n} P(X_i|Y=1) + P(Y=0)\prod_{i=1}^{n} P(X_i|Y=0)}$$

$$P(Y=1|X) = \frac{P(Y=1)\prod_{i=1}^{n} \phi_{i1}^{X_i}(1-\phi_{i1})^{1-X_i}}{P(Y=1)\prod_{i=1}^{n} \phi_{i1}^{X_i}(1-\phi_{i1})^{1-X_i} + P(Y=0)\prod_{i=1}^{n} \phi_{i0}^{X_i}(1-\phi_{i0})^{1-X_i}}$$

$$P(Y=1|X) = \frac{1}{1 + \frac{P(Y=0)}{P(Y=1)}\prod_{i=1}^{n} (\frac{\phi_{i0}}{\phi_{i1}})^{X_i}(\frac{1-\phi_{i0}}{1-\phi_{i1}})^{1-X_i}}$$

$$P(Y=1|X) = \frac{1}{1 + exp\left(\ln(\frac{P(Y=0)}{P(Y=1)}) + \sum_{i=1}^{n}(X_i \ln(\frac{\phi_{i0}}{\phi_{i1}}) + \ln(\frac{1-\phi_{i0}}{1-\phi_{i1}}) - X_i \ln\frac{1-\phi_{i0}}{1-\phi_{i1}})\right)}$$

Assuming $P(Y = 1)$ to be some value π.

$$P(Y=1|X) = \frac{1}{1 + exp\left(\left[\ln(\frac{1-\pi}{\pi}) + \sum_{i=1}^{n} \ln(\frac{1-\phi_{i0}}{1-\phi_{i1}})\right] + \sum_{i=1}^{n}\left[X_i \ln(\frac{\phi_{i0}(1-\phi_{i1})}{\phi_{i1}(1-\phi_{i0})})\right]\right)} = \frac{1}{1 + exp\left(b + \sum_{i=1}^{n} w_i X_i\right)}$$

where w_i and b are defined as: $w_i = \ln(\frac{\phi_{i0}(1-\phi_{i1})}{\phi_{i1}(1-\phi_{i0})})$ and $b = \ln(\frac{1-\pi}{\pi}) + \sum_{i=1}^{n} \ln(\frac{1-\phi_{i0}}{1-\phi_{i1}})$

Similarly,

$$P(Y = 0|X) = 1 - P(Y = 1|X) = 1 - \frac{1}{1 + exp\left(\left[\ln(\frac{1-\pi}{\pi}) + \sum_{i=1}^{n} \ln(\frac{1-\phi_{i0}}{1-\phi_{i1}})\right] + \sum_{i=1}^{n}\left[X_i \ln(\frac{\phi_{i0}(1-\phi_{i1})}{\phi_{i1}(1-\phi_{i0})})\right]\right)}$$

$$\therefore P(Y = 0|X) = \frac{exp\left(b + \sum_{i=1}^{n} w_i X_i\right)}{1 + exp\left(b + \sum_{i=1}^{n} w_i X_i\right)}$$

7. (3 points) **[GROUP]** In a Naive Bayes problem, suppose we are trying to compute $P(Y \mid X_1, X_2, X_3, X_4)$. Furthermore, suppose X_2 and X_3 are identical (i.e., X_3 is just a copy of X_2). Which of the following are true in this case?

Select all that apply:

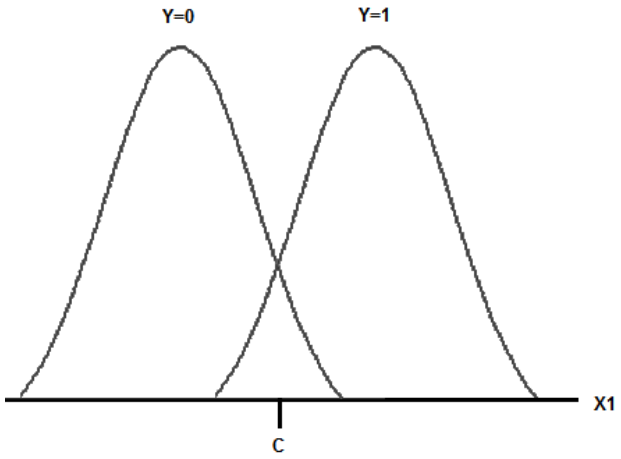
- ■ Naive Bayes will learn identical parameter values for $P(X_2|Y)$ and $P(X_3|Y)$.
- ■ Naive Bayes will output probabilities $P(Y|X_1, X_2, X_3, X_4)$ that are closer to 0 and 1 than they would be if we removed the feature corresponding to X_3.
- □ There is not enough information to determine the change in the output $P(Y|X_1, X_2, X_3, X_4)$.
- □ None of the above

8. (4 points) **[GROUP]** Gaussian Naive 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:

$Y \sim$ Bernoulli(ϕ)
$X_1 \sim$ Gaussian(μ_y, σ_y^2)

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, of course, can be described by a rule of the form "if $X_1 > c$ then $Y = 1$, else $Y = 0$", where c is a real-valued threshold (see diagram provided). Is it possible in this simple one-dimensional case to construct a Gaussian Naive Bayes classifier with a decision boundary that cannot be expressed by a rule in the above form)?

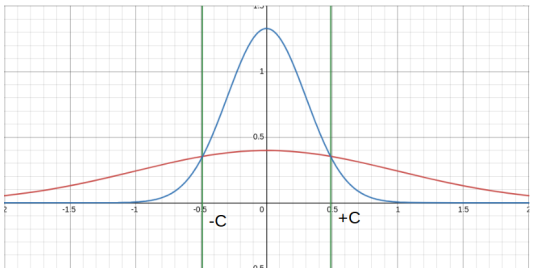


Select all that apply:

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

Draw the corresponding Gaussians and the decision boundaries:

Your answer



In the above, blue represents $Y = 1$ and red represents $Y = 0$.
The decision boundary:

$$Y = \begin{cases} 1, & \text{if } -C < X1 < C \\ 0 & \text{else} \end{cases}$$

5 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