Chris Piech
CS109

Fall Quarter
Dec 14th, 2017

CS109 Final Exam

This is a closed calculator/computer exam. You are, however, allowed to use notes in the exam. The last page of the exam is a Standard Normal Table, in case you need it. The penultimate page has some common random variables. You have 3 hours (180 minutes) to take the exam. The exam is 180 points, meant to roughly correspond to one point per minute of the exam. You may want to use the point allocation for each problem as an indicator for pacing yourself on the exam.

In the event of an incorrect answer, any explanation you provide of how you obtained your answer can potentially allow us to give you partial credit for a problem. For example, describe the distributions and parameter values you used, where appropriate. It is fine for your answers to include summations, products, factorials, exponentials, and combinations, unless the question specifically asks for a numeric quantity or closed form. Where numeric answers are required, the use of fractions is fine.

I acknowledge and accept the letter and spirit of the honor code. I pledge to write more neatly than I have in my entire life:

Signature: ______________________________

Family Name (print): ______________________________

Given Name (print): ______________________________

Email (preferably your gradescope email): ______________________

1 Missing Not at Random [34 points]

You collect data on whether or not people intend to vote for Ayesha, a candidate in an upcoming election. You send an electronic poll to 100 randomly chosen people. You assume all 100 responses are IID.

User Response	Count
Responded that they will vote for Ayesha	40
Responded that they will not vote for Ayesha	45
Did not respond	15

Let A be the event that a person says they will vote for Ayesha. Let M be the event that a user did not respond to the poll. We are interested in estimating $P(A)$, however that is hard given the 15 users who did not respond.

a. (3 points) What is the fraction of users that responded to the poll, $P(M^C)$?

b. (5 points) What is the probability that a user said they will vote for Ayesha, given that they responded to the poll $P(A|M^C)$?

c. (8 points) You estimate that $P(M|A) = \frac{1}{5}$ and that $P(M|A^C) = \frac{1}{10}$. What is $P(A|M)$? You may leave your answer in terms of $P(A)$. To make your future life easier, simplify your answer as much as possible. For this problem, do not use probabilities you calculated in earlier parts.

d. (8 points) Using the same assumption as part (c), calculate $P(A)$. You may use a function `root(a, b, c)` which returns the value of x such that $ax^2 + bx + c = 0$.

e. (10 points) Prove that $P(A) = P(A|M^C)$ if people who did not respond are just as likely to vote for Ayesha as those who did: $P(A|M) = P(A|M^C)$.

2 Occupy Mt Gox [24 points]

Modern nations have such high disparity in incomes that we need to use special fat-tailed random variables to represent country wide distributions. Enter the ExamPareto distribution, a simplified version of the Pareto income distribution. $X \sim \text{ExamPareto}(\alpha)$ represents a countries distribution of incomes—in Bitcoins (B):

Pdf: $$f_X(x) = \begin{cases} \alpha(1+x)^{-(\alpha+1)} & x \geq 0 \\ 0 & \text{else} \end{cases}$$

Mean: $$E[X] = \frac{1}{\alpha-1}$$

Cdf: $$F_X(x) = \begin{cases} 1-(1+x)^{-\alpha} & x \geq 0 \\ 0 & else \end{cases}$$

Variance: $$\text{Var}(X) = \frac{\alpha}{(\alpha-1)^2(\alpha-2)}$$

a. (4 points) Let $X \sim \text{ExamPareto}(\alpha = 2)$. What is the probability that X is greater than 4 B?

b. (6 points) Let $X \sim \text{ExamPareto}(\alpha = 3)$. If we sample 100 IID individual incomes from X, approximate the probability that the sum of their incomes is greater than 55 B?

c. (4 points) You have samples of n IID incomes from a country $(x_1, x_2, \ldots, x_n)$. Write an equation for the log likelihood of the n samples.

d. (10 points) Write pseudocode that can estimate the parameter α from the n samples. You may use any parameter estimation technique we talked about in class. Make sure to work out any necessary math first.

3 Timing Attack [33 points]

In this problem we are going to show you how to crack a password in linear time, by measuring how long the password check takes to execute (see code below). Assume that our server takes T ms to execute any line in the code where $T \sim N(\mu = 5, \sigma^2 = 0.5)$. The amount of time taken to execute a line is always independent of other values of T.

```
# An insecure string comparison
def stringEquals(guess, password):
    nGuess = len(guess)
    nPassword = len(password)
    if nGuess != nPassword:
        return False                  # 4 lines executed to get here
    for i in range(nGuess):
        if guess[i] != password[i]:
            return False              # 6 + 2i lines executed to get here
    return True                       # 5 + 2n lines executed to get here
```

On our site all passwords are length 5 through 10 (inclusive) and are composed of lower case letters only. A hacker is trying to crack the root password which is "gobayes" by carefully measuring how long we take to tell them that her guesses are incorrect. All answers to this question may be left in terms of the Normal CDF function ϕ.

a. (4 points) First, lets say the hacker wanted to brute force and try all passwords. How many passwords are there? Recall that all passwords are length 5 through 10 and composed of lower case letters only.

b. (5 points) What is the distribution of time that it takes our server to execute k lines of code? Recall that each line independently takes $T \sim N(\mu = 5, \sigma^2 = 0.5)$ ms.

c. (7 points) First the hacker needs to find out the length of the password. What is the probability that the time taken to test a guess of correct length (server executes 6 lines) is longer than the time taken to test a guess of an incorrect length (server executes 4 lines)? Assume that the first letter of the guess does not match the first letter of the password. Hint $P(A > B)$ is the same as $P(A - B > 0)$.

d. (5 points) Let p be your answer to part (c). The hacker first tries a guess of length 5, then a guess of length 6, and so on up until a guess of length 10. What is the probability that the guess of length 7, which is the correct length (server executes 6 lines), takes longer than all the other guesses of incorrect length (where the server executes 4 lines)?

e. (8 points) Now that our hacker knows the length of the password, to get the actual string she is going to try and figure out each letter one at a time, starting with the first letter. The hacker tries the string "aaaaaaa" and it takes 27s. Based on this timing, how much more probable is it that first character matched (server executes 8 lines) than the first character did not match (server executes 6 lines)? Assume that all letters in the alphabet are equally likely to be the first letter.

f. (3 points) If it takes the hacker 5 guesses to find the length of the password, and 26 guesses per letter to crack the password string, how many attempts must she make to crack our password, "gobayes"? Yikes!

4 Learning While Helping [30 points]

You are designing a randomized algorithm that delivers one of two new drugs to patients who come to your clinic—each patient can only receive one of the drugs. Initially you know nothing about the effectiveness of the two drugs. You are simultaneously trying to learn which drug is the best and, at the same time, cure the maximum number of people. To do so we will use the Thompson Sampling Algorithm.

Thompson Sampling Algorithm: The algorithm we are going to use is based on sound probability. For each drug we maintain a Beta distribution to represent the drug's probability of being successful. Initially we assume that drug i has a probability of success $\theta_i \sim Beta(1,1)$.

When choosing which drug to give to the next patient we sample a value from each Beta and select the drug with the largest sampled value. We administer the drug, observe if the patient was cured, and update the Beta that represents our belief about the probability of the drug being successful. Repeat for the next patient.

a. (4 points) Imagine you try the first drug on 7 patients. It cures 5 patients and has no effect on 2. What is your updated belief about the drug's probability of success, θ_1? Your answer should be a Beta.

b. (4 points) After running Thomspons' Algorithm 50 times, you end up with the following Beta distribution for the second drug: $\theta_2 \sim \text{Beta}(16, 14)$. What is the variance of a value sampled from the second drug's Beta?

Method	Description
`V = sampleBeta(a, b)`	Returns a real number value in the range [0, 1] with probability defined by a PDF of a Beta with parameters a and b.
`R = giveDrug(i)`	Gives drug i to the next patient. Returns a True if the drug was successful in curing the patient or False if it was not. Throws an error if $i \notin \{1, 2\}$.
`I = argmax(list)`	Returns the index of the largest value in the list.

c. (10 points) Write pseudocode to administer either of the two drugs to 100 patients using Thompson's Sampling Algorithm. Use functions from the table above. Your code should execute `giveDrug` 100 times.

d. (4 points) After running Thomspons' Algorithm 100 times, you end up with the following Beta distributions:
   $\theta_1 \sim \text{Beta}(11, 11)$,
   $\theta_2 \sim \text{Beta}(76, 6)$
   What is the expected probability of success for each drug?

e. (8 points) Write pseudocode to compute the p-value for the difference in expected probability of success between drug2 and drug1 observed from the 100 samples in part (d). Ie, under the null hypothesis assumption that all 100 samples are identically distributed, calculate the probability of a difference in expected probabilities that is greater than or equal to the one observed. Describe, don't implement any helper methods you want to use.

5 Expected Maximum Uniform [12 points]

Let $X \sim \text{Uni}(0,1)$ and $Y \sim \text{Uni}(0,1)$ be independent random variables. What is $\text{E}[\texttt{max}(X, Y)]$? Gain most of the points by writing an expression that you could solve using Wolfram Alpha. Provide a numerical answer for full credit.

6 Gaussian Naïve Bayes [16 points]

The version of Naïve Bayes that we used in class worked great when the feature values were all binary. If instead they are continuous, we are going to have to rethink how we estimate of the probability of the ith feature given the label, $P(X_i|Y)$. The ubiquitous solution is to make the Gaussian Input Assumption that:

If $Y = 0$, then $X_i \sim N(\mu_{i,0}, \sigma^2_{i,0})$

If $Y = 1$, then $X_i \sim N(\mu_{i,1}, \sigma^2_{i,1})$

For each feature, there are 4 parameters (mean and variance for both class labels). There is a final parameter, p, which is the estimate of $P(Y = 1)$. Assume that you have trained on data with two input features and have already estimated all 9 parameter values:

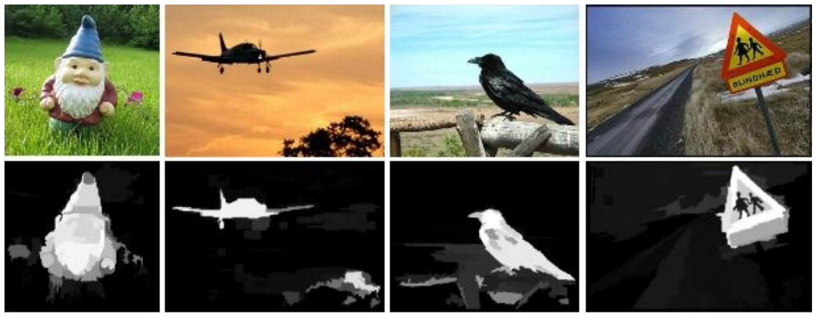
Feature i	$\mu_{i,0}$	$\mu_{i,1}$	$\sigma^2_{i,0}$	$\sigma^2_{i,1}$
1	5	0	1	1
2	0	3	1	4

$$p = 0.6$$

Write an inequality to predict whether $Y = 1$ for input $[X_1 = 5, X_2 = 3]$. Use the Naïve Bayes assumption and the Gaussian Input Assumption. Your expression must be in terms of the learned parameters (either using numbers or symbols is fine):

7 The Most Important Features [32 points]

Let's explore saliency, a measure of how important a feature is for classification. We define the saliency of the ith input feature for a given example $(\mathbf{x}, y)$ to be the absolute value of the partial derivative of the log likelihood of the sample prediction, with respect to that input feature $\left|\frac{\partial LL}{\partial x_i}\right|$. In the images below, we show both input images and the corresponding saliency of the input features (in this case, input features are pixels):



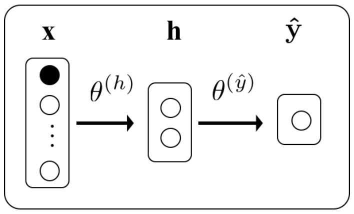
First consider a trained logistic regression classifier with weights $\boldsymbol{\theta}$. Like the logistic regression classifier that you wrote in your homework it predicts binary class labels. In this question we allow the values of x to be real numbers, which doesn't change the algorithm (neither training nor testing).

a. (4 points) What is the Log Likelihood of a single training example $(\mathbf{x}, y)$ for a logistic regression classifier?

b. (8 points) Calculate is the saliency of a single feature (x_i) in a training example $(\mathbf{x}, y)$.

c. (5 points) Show that the ratio of saliency for features i and j is the ratio of the absolute value of their weights $\frac{|\theta_i|}{|\theta_j|}$.

d. (10 points) Saliency is much more interesting for deep learning networks. Consider a simple deep learning network with two hidden neurons that predicts $P(Y=1|\mathbf{X}=\mathbf{x}) \approx \hat{y}$:



$$h_1 = \sigma\Big(\sum_{i=1}^{n} \theta_{i,1}^{(h)} x_i\Big) \qquad h_2 = \sigma\Big(\sum_{i=1}^{n} \theta_{i,2}^{(h)} x_i\Big) \qquad \hat{y} = \sigma\Big(\theta_1^{(y)} h_1 + \theta_2^{(y)} h_2\Big)$$

Calculate the saliency of a single feature (x_i) in a training example ($\mathbf{x}, y$).

e. (5 points) Weights are not invariant to units of measurement. Imagine your logistic regression classifier has weight values: $[\theta_0 = 10, \theta_1 = -4, \theta_2 = 8]$. What values do you expect for the weights in a newly trained model if we change the units of the last feature (x_2) to be centimeters instead of meters? That means all the values x_2 will be 100x times as large.

$\theta_0 =$
$\theta_1 =$
$\theta_2 =$

[End of exam]

That’s all folks! Missing data requires probability theory to understand. Income distributions give us a way to measure inequality. Timing attacks are a real thing... watch your code. Thompson Sampling is a hot new algorithm for N-Armed-Bandit questions, a key part of MCTS, the algorithm used in AlphaGoZero. Gaussian Naïve bayes is good times for continuous inputs and saliency is a new tool for understanding deep neural nets. Thank you all for the wonderful quarter.

.

Continuous Random Variable Reference

Uniform
All intervals in the range (of the same length) are equally probable.

Notation	$X \sim \text{Uni}(\alpha, \beta)$
Range(X):	$x \in \mathbb{R},\ a \leq x \leq b$
pdf:	$f(X = x) = \frac{1}{b-a}$
$E[X]$:	$\frac{1}{2}(a+b)$
Var(X):	$\frac{1}{12}(b-a)^2$

Normal (Gaussian)
Often used to represent real-valued random variables whose distributions are not known.

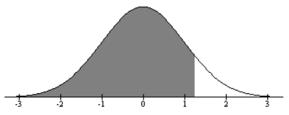
Notation:	$X \sim N(\mu, \sigma^2)$	Support:	$x \in \mathbb{R}$
PDF:	$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$	CDF:	$F(x) = \Phi(\frac{x-\mu}{\sigma})$
$E[X]$:	μ	Var(X):	σ^2

Beta
Often used to represent of belief of a probability.

Notation:	$X \sim \text{Beta}(a, b)$	Support:	$x \in \mathbb{R},\ 0 \leq x \leq 1$
PDF:	$f(x) = \frac{1}{B(a,b)} x^{(a-1)}(1-x)^{(b-1)}$		
$E[X]$:	$\frac{a}{a+b}$	Mode:	$\frac{a-1}{a+b-2}$
Var(X):	$\frac{ab}{(a+b)^2(a+b+1)}$		

Standard Normal Table

An entry in the table is the area under the curve to the left of z, $P(Z \leq z) = \Phi(z)$.



Z	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0.0	0.5000	0.5040	0.5080	0.5120	0.5160	0.5199	0.5239	0.5279	0.5319	0.5359
0.1	0.5398	0.5438	0.5478	0.5517	0.5557	0.5596	0.5636	0.5675	0.5714	0.5753
0.2	0.5793	0.5832	0.5871	0.5910	0.5948	0.5987	0.6026	0.6064	0.6103	0.6141
0.3	0.6179	0.6217	0.6255	0.6293	0.6331	0.6368	0.6406	0.6443	0.6480	0.6517
0.4	0.6554	0.6591	0.6628	0.6664	0.6700	0.6736	0.6772	0.6808	0.6844	0.6879
0.5	0.6915	0.6950	0.6985	0.7019	0.7054	0.7088	0.7123	0.7157	0.7190	0.7224
0.6	0.7257	0.7291	0.7324	0.7357	0.7389	0.7422	0.7454	0.7486	0.7517	0.7549
0.7	0.7580	0.7611	0.7642	0.7673	0.7703	0.7734	0.7764	0.7793	0.7823	0.7852
0.8	0.7881	0.7910	0.7939	0.7967	0.7995	0.8023	0.8051	0.8078	0.8106	0.8133
0.9	0.8159	0.8186	0.8212	0.8238	0.8264	0.8289	0.8315	0.8340	0.8365	0.8389
1.0	0.8413	0.8438	0.8461	0.8485	0.8508	0.8531	0.8554	0.8577	0.8599	0.8621
1.1	0.8643	0.8665	0.8686	0.8708	0.8729	0.8749	0.8770	0.8790	0.8810	0.8830
1.2	0.8849	0.8869	0.8888	0.8906	0.8925	0.8943	0.8962	0.8980	0.8997	0.9015
1.3	0.9032	0.9049	0.9066	0.9082	0.9099	0.9115	0.9131	0.9147	0.9162	0.9177
1.4	0.9192	0.9207	0.9222	0.9236	0.9251	0.9265	0.9279	0.9292	0.9306	0.9319
1.5	0.9332	0.9345	0.9357	0.9370	0.9382	0.9394	0.9406	0.9418	0.9429	0.9441
1.6	0.9452	0.9463	0.9474	0.9484	0.9495	0.9505	0.9515	0.9525	0.9535	0.9545
1.7	0.9554	0.9564	0.9573	0.9582	0.9591	0.9599	0.9608	0.9616	0.9625	0.9633
1.8	0.9641	0.9649	0.9656	0.9664	0.9671	0.9678	0.9686	0.9693	0.9699	0.9706
1.9	0.9713	0.9719	0.9726	0.9732	0.9738	0.9744	0.9750	0.9756	0.9761	0.9767
2.0	0.9772	0.9778	0.9783	0.9788	0.9793	0.9798	0.9803	0.9808	0.9812	0.9817
2.1	0.9821	0.9826	0.9830	0.9834	0.9838	0.9842	0.9846	0.9850	0.9854	0.9857
2.2	0.9861	0.9864	0.9868	0.9871	0.9875	0.9878	0.9881	0.9884	0.9887	0.9890
2.3	0.9893	0.9896	0.9898	0.9901	0.9904	0.9906	0.9909	0.9911	0.9913	0.9916
2.4	0.9918	0.9920	0.9922	0.9925	0.9927	0.9929	0.9931	0.9932	0.9934	0.9936
2.5	0.9938	0.9940	0.9941	0.9943	0.9945	0.9946	0.9948	0.9949	0.9951	0.9952
2.6	0.9953	0.9955	0.9956	0.9957	0.9959	0.9960	0.9961	0.9962	0.9963	0.9964
2.7	0.9965	0.9966	0.9967	0.9968	0.9969	0.9970	0.9971	0.9972	0.9973	0.9974
2.8	0.9974	0.9975	0.9976	0.9977	0.9977	0.9978	0.9979	0.9979	0.9980	0.9981
2.9	0.9981	0.9982	0.9982	0.9983	0.9984	0.9984	0.9985	0.9985	0.9986	0.9986
3.0	0.9987	0.9987	0.9987	0.9988	0.9988	0.9989	0.9989	0.9989	0.9990	0.9990
3.1	0.9990	0.9991	0.9991	0.9991	0.9992	0.9992	0.9992	0.9992	0.9993	0.9993
3.2	0.9993	0.9993	0.9994	0.9994	0.9994	0.9994	0.9994	0.9995	0.9995	0.9995
3.3	0.9995	0.9995	0.9995	0.9996	0.9996	0.9996	0.9996	0.9996	0.9996	0.9997
3.4	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9998
3.5	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998

Filler