

Section 9: Quiz #3 Review

1. Continuous Joint Distributions (courtesy of Oishi Banerjee)

- a. Let X , Y , and Z be independent Normal variables with means of $\mu_X = 4$, $\mu_Y = 5$, and $\mu_Z = 6$ and variances $\sigma_X^2 = 16$, $\sigma_Y^2 = 25$, and $\sigma_Z^2 = 36$. If we assume $A = X + Y$ and $B = Y + Z$ are each sums of independent Normal variables, then what is the joint distribution of A and B ? Restated, what is their Bivariate Normal distribution?
- b. Suppose hundreds of thousands (that is, a sufficiently large number) of student scores on a 150-question exam are distributed according to the following random variable:

$$R = \sum_{i=1}^{50} M_i + 0.5 \sum_{j=1}^{100} W_j \quad (1)$$

Each of the M_i are independent and identically distributed (IID) Beta random variables—yes, the questions are scored on a continuous scale from 0 to 1—and the W_j are separate IID Beta random variables, where all W_j are independent of all M_i . The Beta parameters are $\alpha_M = 10$, $\beta_M = 2$, $\alpha_W = 8$, and $\beta_W = 4$. If we sample 100 student scores R_1, \dots, R_n IID according to the distribution of R above, what is the distribution of the sample mean \bar{R} ?

2. Bootstrapping and Null Hypotheses (courtesy of Oishi Banerjee)

While testing the efficacy of a new drug, Skylar Pharmaceuticals has collected 1000 data samples. Most of the samples came from patients who were treated with the drug, but the rest came from patients who received a placebo. Skylar observed that the sample mean blood pressure in the treated group was 80, while the sample mean blood pressure in the placebo group was 86. To demonstrate the difference is statistically significant, Skylar implemented the following to produce a p-value.

Data scientists at Skylar wrote the following bootstrapping code to arrive at a p-value to suggest the difference isn't statistically significant. Unfortunately (or fortunately, depending on your point of view), their code is not right.

```
import numpy as np

def resample(whole, num_samples):
    return np.random.choice(whole, num_samples, replace=True)

# list_treat is an ordinary 1-d numpy array
# it contains all the diastolic blood pressures of
# each patient who was treated
```

```

# list_placebo is an ordinary 1-d numpy array
# it contains the diastolic blood pressures of each
# patient who received a placebo
def pvalue(list_treat, list_placebo):
    # np.concatenate will make a 1000-element array #containing the
    # elements of both list_treat and list_placebo
    whole = np.concatenate([list_treat, list_placebo])
    threshold = np.mean(list_treat) - np.mean(list_placebo)
    counter, num_trials = 0, 100000
    for trial in range(num_trials):
        sample_treat = resample(list_treat, 500)
        sample_placebo = resample(list_placebo, 500)
        mean_treat = np.mean(sample_treat)
        mean_placebo = np.mean(sample_placebo)
        new_diff = np.abs(mean_treat - mean_placebo)
        if new_diff == threshold: counter += 1
    return counter/num_trials

```

Point out the algorithmic errors. Be clear what Skylar should do instead, explaining why each change you would make is necessary for correct bootstrapping.

3. Naïve Bayes (courtesy of David Varodayan and Lisa)

Suppose we observe two discrete input variables X_1 and X_2 and want to predict a single binary output variable Y (which can have values 0 or 1). We know that the functional forms for the input variables are $X_1 \sim \text{Poi}(\lambda)$ and $X_2 \sim \text{Ber}(p)$, but we are not given the values of the parameters λ or p . We are, however, given a dataset of 9 training instances (shown at right.)

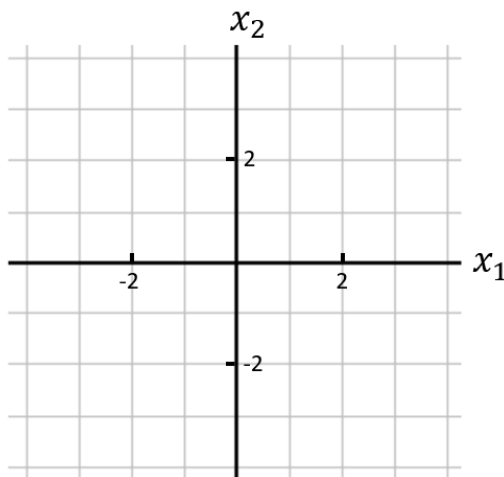
X_1	X_2	Y		X_1	X_2	Y
1	1	0		3	1	1
3	0	0		5	0	1
7	1	0		5	1	1
9	0	0		5	1	1
				7	1	1

- a. Use Maximum Likelihood Estimation to estimate the parameters λ and p in the case where $Y = 0$ as well as the case $Y = 1$. You should have four parameter estimates: λ_0 and p_0 for when $Y = 0$, and λ_1 and p_1 for when $Y = 1$.

- b. Use Maximum Likelihood Estimation to estimate the probability $P(Y = 1)$.

- c. You observe the following testing instance: $(X_1, X_2) = (2, 0)$. Using the Naïve Bayes assumption, predict the output Y for the testing instance. For this problem, showing how you computed your prediction is worth more points than the final answer.

4. **Logistic regression (courtesy of David Varodayan)**



The two parts of this problem are unrelated.

- a. **Prediction.** Suppose you have trained a logistic regression classifier that accepts as input a data point (x_1, x_2) and predicts a class label \hat{Y} . The parameters of the model are $(\theta_0, \theta_1, \theta_2) = (2, 2, -1)$. On the axes, draw the decision boundary $\theta^T \mathbf{x} = 0$ and clearly mark which side of the boundary predicts $\hat{Y} = 0$ and which side predicts $\hat{Y} = 1$.
- b. **Training.** The logistic regression parameter update equation is

$$\theta_j^{\text{new}} = \theta_j^{\text{old}} + \eta \sum_{i=1}^n \left[y^{(i)} - \sigma \left(\theta^{\text{old}T} \mathbf{x}^{(i)} \right) \right] x_j^{(i)}$$

Your training set consists of two data points $(x_1^{(1)}, y^{(1)}) = (1, 1)$ and $(x_1^{(2)}, y^{(2)}) = (-1, 0)$. Given $(\theta_0^{\text{old}}, \theta_1^{\text{old}}) = (0, 0)$ and $\eta = 0.1$, find $(\theta_0^{\text{new}}, \theta_1^{\text{new}})$.