BE 562: Problem Set 1 Solutions

Problem 1. Axioms of Probability. (10 pts)
Using the axioms of probability we learned in the lectures, prove the following:

a) $P(A^c) = 1 - P(A)$

b) $P(A \cup B \cup C) = P(A) + P(B) + P(C) - P(A \cap B) - P(B \cap C) - P(A \cap C) + P(A \cap B \cap C)$

Axiom I. For every event $A \subseteq S$, $P(A) \geq 0$.
Axiom II. $P(S) = 1$
Axiom III. If $A_1, A_2, ...A_n$ are events and $A_i \cap A_j = \emptyset$ for all $i \neq j$, then

$$P(S) = P(A_1 \cup A_2 \cup ... \cup A_n) = \sum_{i=1}^{n} P(A_i)$$

a) We know that $S = A \cup A^c$ and $A \cap A^c = \emptyset$.

From Axiom 3: $P(S) = P(A) + P(A^c)$

From Axiom 2: $P(S) = 1$

Therefore,

$$P(S) = 1 = P(A) + P(A^c)$$
$$P(A^c) = 1 - P(A)$$

b) Let $D = B \cup C$ so that

$$\begin{aligned} P(A \cup B \cup C) &= P(A \cup D) \\ &= P(A) + P(D) - P(A \cap D) \\ &= P(A) + P(B \cup C) - P(A \cap (B \cup C)) \\ &= P(A) + P(B) + P(C) - P(B \cap C) - P(A \cap (B \cup C)) \end{aligned}$$

The Distributive law gives that

$$A \cup (B \cap C) = (A \cap B) \cup (A \cap C)$$

Thus,

$$P(A \cup (B \cap C)) = P(A \cap B) + P(A \cap C) - P((A \cap B) \cap (A \cap C))$$

Since $(A \cap B) \cap (A \cap C) = A \cap B \cap C$

Then,

$$\begin{aligned} P(A \cup B \cup C) = P(A) + P(B) + P(C) \\ - P(B \cap C) - P(A \cap B) + P(A \cap C) \\ - P(A \cap B \cap A \cap C) \end{aligned}$$

Problem 2. Probability. (5 pts)
The chance of being infected by the Zika virus depends on countries visited. The chance of infection in Laos is 40% and 80% in Venezuela and 20% in Australia. The chance of complications for these areas are 9%, 11%, and 13%, respectively. If a traveler spent equal time in all three countries, what is the chance that complications, which occurred were due to time in Laos?

$P(L_i) = 0.4$
$P(V_i) = 0.8$
$P(A_i) = 0.2$

$P(C|L_i) = 0.09$
$P(C|V_i) = 0.11$
$P(C|A_i) = 0.13$

Find $P(L_i|C)$

$$P(L_i|C) = \frac{P(L_i, C)}{P(C)} = \frac{P(C|L_i)P(L_i}{P(C)}$$

where

$$\begin{aligned} P(C) &= \sum_{all\ countries} P(C, countries) \\ &= P(C|L_i)P(L_i) + P(C|V_i)P(V_i) + P(C|A_i)P(A_i) \\ &= (0.09)(0.4) + (0.11)(0.8) + (0.13)(0.2) = 0.15 \end{aligned}$$

$$\begin{aligned} P(L_i|C) &= \frac{P(C|L_i)P(L_i)}{P(C)} \\ &= \frac{(0.09)(0.4)}{0.15} = 0.24 \end{aligned}$$

Problem 3. Probability. (10 pts)
A bin contains three types of influenza A vaccines: H1N1, H3N2 and H10N7. The probability that the H1N1 vaccine will last one season is 0.5, with the corresponding probabilities of the H3N2 and H10N7 vaccines being 0.4 and 0.3, respectively. Suppose that 20% of the vaccines in the bin are for H1N1, 70% for H3N2 and 10% fir H10N7.

a) What is the probability that a vaccine chosen at random will last one season?

b) Given that a vaccine lasted an entire season, what is the conditional probability that it was for H3N2?

$P(H1N1) = 0.2$
$P(H3N2) = 0.7$
$P(H10N7) = 0.1$

$P(Last|H1N1) = 0.5$
$P(Last|H3N2) = 0.4$
$P(Last|H10N7) = 0.3$

a) Find P(Last)

$$P(Last) = \sum_{all\ vaccines} P(Last|Vaccine)P(Vaccine)$$
$$= P(Last|H1N1)P(H1N1) + P(Last|H3N2)P(H3N2) + P(Last|H10N7)P(H10N7)$$
$$= (0.5)(0.2) + (0.4)(0.7) + (0.3)(0.1) = 0.41$$

b) Find P(H3N2|Last)

$$P(H3N2|Last) = \frac{P(Last|H3N2)P(H3N2)}{P(Last)}$$
$$= \frac{(0.4)(0.7)}{0.41} = 0.683$$

Problem 4. Probability. (5 pts)
Show that if $P(A|B) < P(A)$, then $P(B|A) < P(B)$.

$$P(A|B) < P(A)$$
$$\frac{P(A|B)}{P(A)} < 1$$
$$\frac{P(A,B)}{P(A)P(B)} < 1$$
$$\frac{P(B|A)}{P(B)} < 1$$
$$P(B|A) < P(B)$$

Problem 5. Bayes' Rule. (10 pts)
Assume that a particular disease has a prevalence of 1% in the population. A company has developed a diagnostic for this disease that is 90% reliable (i.e. it detects 90% of true cases), but has a false positive rate of 5%.

a) If a person is tested positive by this test, what are the odds that this person has the disease?

b) What is the probability the person does not have the disease?

$P(D) = 0.01$ $\qquad$ $P(+|D) = 0.9$
$P(D^c) = 1 - P(D) = 0.99$ $\qquad$ $P(+|D^c) = 0.05$

a) Find the odds that a person has the disease given that they tested positive, $\frac{P(D|+)}{P(D^c|+)}$.
(Odds that A occurs is defined as the probability that A occurs divided by the probability

that A does not occur.)

$$\frac{P(D|+)}{P(D^c|+)} = \frac{\left(\frac{P(+|D)P(D)}{P(+)}\right)}{\left(\frac{P(+|D^c)P(D^c)}{P(+)}\right)}$$
$$= \frac{P(+|D)P(D)}{P(+|D^c)P(D^c)}$$
$$= \frac{(0.9)(0.01)}{(0.05)(0.99)} = 0.182$$

b) Find $P(D^c|+)$

$$P(D^c|+) = \frac{P(+|D^c)P(D^c)}{P(+)}$$

$$P(+) = P(+, D) + P(+, D^c)$$
$$= P(+|D)P(D) + P(+|D^c)P(D^c)$$
$$= (0.9)(0.01) + (0.05)(0.99) = 0.0585$$

$$P(D^c|+) = \frac{P(+|D^c)P(D^c)}{P(+)} = \frac{(0.05)(0.99)}{0.0585} = 0.846$$

Problem 6. Binomial Hypothesis Testing. (10 pts)
A study claims that a particular single nucleotide polymorphism (SNP) will cause baldness in 70% of the men that carry the mutation. One scientific group thinks this is too high, so they decide to genotype 21 men for this SNP. They find that out of the 15 men that have this SNP, 5 are bald. Can we still conclude, with a significance = 0.05, that this particular SNP will result in baldness in its carriers 70% of the time? (Hint: do a left tailed test) Show all your work.

H_0: 70% of men with this SNP will go bald.
H_a: Less than 70% of the men with this SNP will go bald.
Statistic Test: left-tailed Binomial Test with $\alpha = 0.05$.

$$P(X \leq x) = \sum_{k=0}^{x} [(\binom{N}{k} p^k (1-p)^{N-k}]$$

$$P(X \leq 5) = \sum_{k=0}^{5} [\binom{15}{k} 0.7^k (1-0.7)^{15-k}]$$
$$= 0.0037$$

Because $P(X \leq 5) < 0.05$ reject the null hypothesis that 70% of men with this particular SNP will go bald.

Problem 7. Expected Value and Linearity. (5 pts)
Show that the expected value operator is linear in that:

a) $E[aX + b] = aE[X] + b$

b) $E[aX + bY] = aE[X] + bE[Y]$

a)

$$\begin{aligned} E[aX + b] &= \sum (ax + b)p(x) \\ &= \sum ax\ p(x) + \sum b\ p(x) \\ &= a \sum x\ p(x) + b \\ &= aE[X] + b \end{aligned}$$

b)

$$\begin{aligned} E[aX + bY] &= \sum (ax + by)p(x, y) \\ &= \sum ax\ p(x) + \sum by\ p(y) \\ &= a \sum x\ p(x) + b \sum y\ p(y) \\ &= aE[X] + bE[Y] \end{aligned}$$

Problem 8. Evolutionary distance and whole-genome duplication. (25 pts)
In this problem, you will implement the Needleman-Wunsch algorithm for pairwise sequence alignment, apply it to the protein-coding sequences of related genes from several mammalian genomes, and use the results to learn about their evolution.

a) On the class web site, we have provided a python skeleton program **ps1-seqalign.py**, which you will complete. We provide a traceback routine, but you will write the code to fill in the score and traceback matrices. The skeleton program specifies a substitution matrix and gap penalty. If you so choose, you may rewrite the program in any programming language. Please submit (1) the portion of the code that you wrote; (2) an optimal alignment the two sequences TACGCAG and AGCTG, and corresponding score matrix F with the optimal path indicated; and (3) the score of the alignment of the human and mouse HoxA13 genes, which we also provide on the web site.

The Hox cluster is a set of genes that are crucial in determining body plan formation during embryo development. They are found in all bilateral animals, in species as distant as the fruit fly. The fruit fly has one Hox cluster, while most vertebrates have four. It is thought that vertebrates have undergone two rounds of whole-genome duplication, giving rise to four Hox clusters from the ancestral one.

In the remainder of this problem, you will use your Needleman-Wunsch alignment program to analyze the sequences of several Hox genes, and estimate the date of the most recent vertebrate whole-genome duplication. In particular, we are interested in using the N-W alignment score as a distance metric between two sequences.

b) Make minor adjustments to your alignment program so that the score it computes can be interpreted as a distance metric. For example, the score of a sequence aligned with itself should be zero, and sequences that are more dissimilar should give a score with a greater magnitude. Describe the changes you made in your work; no code is necessary.

c) Apply your modified program to compute a distance between the human HoxA13 gene and the mouse HoxA13 gene. The fossil record shows that human and mouse diverged about 70 million years ago.

d) The modern mammalian genes HoxA13 and HoxD13 arose from a single ancestral gene by whole-genome duplication, long before the human-mouse divergence. We provide the sequences of the human and mouse HoxD13 genes on the web site. Use your distance metric and your results from part (c) to estimate the date of the whole-genome duplication that gave rise to HoxA13 and HoxD13. Make sure to state the assumptions underlying your estimate.

a) Alignment:

```
T A C G C A G
  |   | |   |
- A - G C T G
```

Score: 2

```
for i in range(1, len(seq1)+1):
  for j in range(1, len(seq2)+1):
    a = F[i-1][j-1]+ subst_matrix[base_idx[seq1[i-1]]][base_idx[seq2[j-1]]] #
                                        match/mismatch
    b = F[i-1][j]-gap_penalty # gap in seq 2
    c = F[i][j-1]-gap_penalty # gap in seq 1

    F[i][j] = max(a,b,c)

    if max(a,b,c) == a:
      TB[i][j] = PTR_BASE
    elif max(a,b,c) == b:
      TB[i][j] = PTR_GAP2
    elif max(a,b,c) == c:
      TB[i][j] = PTR_GAP1
```

Note: Questions 8b-d answers will vary depending on what your adjustment is to your alignment program and the distance you get to show divergence.

Problem 9. Dynamic programming for multiple sequence alignment. (15 pts)
Give a dynamic programming recurrence for computing the optimal semi-global alignment

of three sequences. You do not need to describe how to fill in the dynamic programming table. Assume that you have a function, $s(i, j, k)$, that will provide the score of aligning three nucleotides and/or gaps.

$$F(i,j,k) = \begin{cases} F(i-1,j-1,k-1) + s(i,j,k) \\ F(i,j,k-1) + s(gap,gap,k) \\ F(i-1,j,k) + s(i,gap,gap) \\ F(i,j-1,k) + s(gap,j,gap) \\ F(i,j-1,k-1) + s(gap,j,k) \\ F(i-1,j,k-1) + s(i,gap,k) \\ F(i-1,j-1,k) + s(i,j,gap) \end{cases}$$

Problem 10. Sequence hashing and dotplot visualization. (20 pts)
As you have seen in class, sequence alignment is a quadratic time algorithm. Full sequence alignment is therefore only feasible for sequences near the length of a single gene. To align larger regions of a genome, heuristic approximations are typically used. In this problem, you will use hashing techniques to guide the alignment of a 1 megabase (1 million nucleotides) region surrounding the HoxA cluster in human (**human-hoxa-region.fa**) and mouse (**mouse-hoxa-region.fa**). You will use dotplots to isualize the performance of various hashing methodologies.

The code provided (**ps1-dotplot.py**) finds all 30-mers in the human that also appear in mouse. On a dotplot, each of these matches is represented as a single dot at (x, y), where x is a coordinate for the beginning of a 30-mer in human and y is a coordinate for the beginning of a matching 30-mer in mouse. We provide a plotting function that will produce dotplot images. The format of the image is determined by the file extension (*.ps, *.png, *.jpg). There is also code for heuristically judging the specificity of the matches (the fraction of matches that occur near the diagonal of the dotplot).

a) Run the script unchanged to generate a dotplot for all exact matching 30-mers. Describe what you see. How many hits are there and what percentage fall near the diagonal? Do you observe any structure in the off diagonal hits? What types of genomic elements could cause such a pattern? Why are matches that are close to the diagonal more likely than off-diagonal matches to represent "correct", or orthologous, alignments?

b) Make the following modifications to the script and report how the plot changes qualitatively and quantitatively (how many hits, what percentage is near the diagonal). Also briefly describe how you implemented each change.

- **i.** Modify the script to find all exact matching 100-mers
- **ii.** Modify the script to find all 60-mers that match every other base
- **iii.** Modify the script to find all 90-mers that match every third base
- **iv.** Modify the script to find all 120-mers that match every fourth base

v. Modify the script to find all 100-mers that allow a mismatch every third base

a) There are 62829 hits with 24.7% on the diagonal.

b) **i.** 1198 hits, 100% on diagonal. Changes needed:

- $kmerlen = 100$

ii. 23933 hits, 38.7% on diagonal. Changes needed:

- $kmerlen = 60$
- $key = seq1[i : i + kmerlen : 2]$
- $key = seq2[i : i + kmerlen : 2]$

iii. 8887 hits, 93.9% on diagonal. Changes needed:

- $kmerlen = 90$
- $key = seq1[i : i + kmerlen : 3]$
- $key = seq2[i : i + kmerlen : 3]$

iv. 6044 hits, 82.1% on diagonal. Changes needed:

- $kmerlen = 120$
- $key = seq1[i : i + kmerlen : 4]$
- $key = seq2[i : i + kmerlen : 4]$

v. 2772 hits, 100% on diagonal OR 2870 hits, 99.97% on diagonal if you skip the first and then every third from there. Changes needed:

- $kmerlen = 100$

There are several ways to solve this problem, but you will need to manipulate the sequence to remove every third base or ignore every third base. (Could use splits, phyisically identifying index...)