Comp 5970 Assignment 1

Haden Stuart

1. Probabilistic Reasoning

a)

V	$p(A = 1 \mid V)$	$p(K = 1 \mid V)$	$p(L = 1 \mid V)$	**prior** $p(V)$
0	2/6	3/6	2/6	1/2
1	4/6	5/6	2/6	1/2

b) For p(V = 1 | A = 0, K = 1, L = 0):

- ⇨ p(V | A, K, L) = p(V | A) * p(V | K) * p(V | L)
- ⇨ p(V | A) = [p(A | V) * p(V) / p(A)] = (2/6) * (1/2) / (1/2) = 1/3
- ⇨ p(V | K) = [p(K | V) * p(V) / p(K)] = (5/6) * (1/2) / (2/3) = 5/8
- ⇨ p(V | L) = [p(L | V) * p(V) / p(L)] = (4/6) * (1/2) / (2/3) = 1/2
- ⇨ p(V = 1 | A = 0, K = 1, L = 0) = (1/3) * (5/8) * (1/2) = **5/48 = 0.104**

For p(V = 0 | A = 0, K = 1, L = 0):

- ⇨ p(V | A, K, L) = p(V | A) * p(V | K) * p(V | L)
- ⇨ p(V | A) = [p(A | V) * p(V) / p(A)] = (4/6) * (1/2) / (1/2) = 2/3
- ⇨ p(V | K) = [p(K | V) * p(V) / p(K)] = (3/6) * (1/2) / (2/3) = 3/8
- ⇨ p(V | L) = [p(L | V) * p(V) / p(L)] = (4/6) * (1/2) / (2/3) = 1/2
- ⇨ p(V = 1 | A = 0, K = 1, L = 0) = (2/3) * (3/8) * (1/2) = **1/8 = 0.125**

Given these results, we can conclude that message M has a higher probability of not having a virus.

c) For p(V = 0 | A = 0, K = 1, L = 0) we get: 1/2 = 0.5
For p(V = 1 | A = 0, K = 1, L = 0) we get: 1/2 = 0.5

These results are not the same that we have calculated in the previous problem because in the previous problem we had to use the probabilities obtained from Bayes' rule to calculate the independent probability whereas in this problem we had to base the solution solely off of the data we were provided.

d) One constraint that we must observe is that the summation of each column must be between 0 and 1 as well.

e) By changing one value of A in the table we could drastically alter the data to either boost or reduce the probability of there being a virus, since the equation to get the result relies on p(V | A).

f) In order to find the probability values for p(A, L, K | V), we would need to specify at least two of the three probability values in the distribution. With these two values given, we would be able to solve for the last value, since all values in a distribution must up sum to 1.

g) The independence assumption would not hold in reality because each distinct probability would not truly be independent since having one of the three possibilities could alter the probability of having another.

2. Maximum Likelihood Estimation

a) So, to start off we know that the log likelihood of the Poisson distribution is:

$$P(X = x) = \frac{u^x e^{-u}}{x!}, u > 0$$

Now let's set the sample to ($x_1 \ldots x_{50}$), which gives us:

$$\sum_{i=1}^{50} x_i = 150$$

Each $x_i \sim Pois(\lambda)$, so the pdf of each single x_i will be:

$$f(x_i \mid \lambda) = \frac{e^{-\lambda}\lambda^{x_i}}{x_i!}$$

Now using this equation, we can calculate the likelihood function:

$$L(\lambda \mid x_1 \ldots x_{50}) = \prod_{i=1}^{50} f(x_i \mid \lambda) \Rightarrow \prod_{i=1}^{50} \frac{e^{-\lambda}\lambda^{x_i}}{x_i!}$$

With this we can now get the log likelihood function:

$$l(\lambda \mid x_1 \ldots x_{50}) = \sum_{i=1}^{50} \log\left(\frac{e^{-\lambda}\lambda^{x_i}}{x_i!}\right) \Rightarrow \sum_{i=1}^{50} (-\lambda + x_i \log(\lambda) - \log(x_i!))$$

Since the last value in the previous equation is a constant with respect to λ we can drop it:

$$l(\lambda \mid x_1 \ldots x_{50}) = \sum_{i=1}^{50} (-\lambda + x_i \log(\lambda)) \Rightarrow -50\lambda + \sum_{i=1}^{50} x_i \log(\lambda)$$

Now just take the derivative with respect to λ:

$$l'(\lambda \mid x_1 \ldots x_{50}) = -50 + \frac{\sum x_i}{\lambda}$$

Set this equal to 0 and solve for λ:

$$-50 + \frac{\sum x_i}{\lambda} = 0 \Rightarrow \lambda = \frac{\sum x_i}{50}$$

Since 50 is our example size, we can just set the denominator to n, meaning max likelihood is:

$$\widehat{\lambda} = \frac{\sum x_i}{n}$$

b) First we must find argmax P(u | D) where D = ($x_1 \ldots x_n$) by using Bayes' rule:

$$\Rightarrow arg\max \frac{p(D \mid u)p(u)}{p(D)}$$

Since the denominator is a constant with respect to u, we can drop it:

$$\Rightarrow arg\max p(D \mid u)p(u)$$

Take the log of the equation above and split into sum of logs:

$$arg\max \left(\log p(D \mid u) + \log p(u)\right)$$

Now we take the derivative and set it equal to 0:

$$\frac{d}{du}\left(\log p(D \mid u) + \log p(u)\right) = 0$$

This computes to:

$$\frac{d}{du}\left(n \log u - u\sum_{i=1}^{n} x_i + \log\left(\lambda e^{\lambda} u\right)\right) \Rightarrow \frac{n}{u} - \sum_{i=1}^{n} x_i + \lambda = 0$$

This leaves us with the maximum posteriori being:

$$\widehat{u} = \frac{n}{\sum_{i=1}^{n} x_i - \lambda}$$

3. Entropy

a) Given the total unique words is N, the maximum value is going to be halfway between the smallest(1) and the largest(N) sample size since at this point we will have the highest uncertainty. The minimum value for this would be on either side of the maximum meaning when the sample size is the smallest(1) and largest(N), since we would have either have an infinitely large vocabulary to choose from or an almost empty vocabulary to choose from.

b) For minimum:

1.) If we are given document D_1 that contains none of the 6 words in our sample, then we are completely certain that we won't choose a word from the list, meaning H(W) = 0.

2.) If we are given document D_2 that only has words from our sample, then we know for sure we will choose a word from our list, meaning H(W) = 0.

For maximum:

1.) If we are given document D_3 that has exactly half of the words from our sample, then we will be almost completely uncertain whether or not we will choose a word from our list, meaning H(W) = 1.

2.) If we are given document D_4 that contains every word from our list along with 6 other random words that are not in our list, then we will once again be almost completely uncertain as to whether we will choose a word from our list or one of the random words not in our list, meaning H(W) = 1.

c) If we are given that H(W) = 0, then we are completely certain that both A_1 and A_2 either have words from our vocabulary or do not. This means that if we were to concatenate the two together, we would still be completely certain that the word would either be in or not be in our vocabulary, meaning that H(W) = 0. If we say that A_1 contains {red, blue, green, yellow} and A_2 contains {orange, purple, red, white}, and we search for the word {red} then we know for sure that before and after the concatenation, the sample will contain the word.

4. Conditional Entropy and Mutual Information

a) The value for the conditional entropy of H(X | X) will be 0 because there is no uncertainty of X given X.

b) If X and Y are independent, then the mutual information I(X;Y) = 0.
The equation for mutual information is H(X) – H(X | Y), but if X and Y are independent then the equation becomes H(X) – H(X), which equals 0.

5. Mutual Information of Words

a) $p(X_A = 0, X_B = 1) = (N_B - N_{AB}) / N$

$p(X_A = 0, X_B = 0) = (N - N_{AB}) / N$

b) For $I(X_{computer}; X_{program})$:

$p(computer, program) * \log_2[p(computer, program) / (p(computer) * p(program))]$

$= (N_{AB}/N) * \log_2[(N_{AB}/N) / (((N_A - N_{AB}) / N) * ((N_B - N_{AB}) / N))]$

$= (349 / 26394) * \log_2[(349 / 26394) / (((1390 - 349)/26394) * ((2370 - 349) / 26394))]$

$= (0.0132) * \log_2[(0.0132) / ((0.0394) * (0.0766))] = 0.0132 * \log_2(4.374) = 0.028$

For $I(X_{computer}; X_{baseball})$:

p(computer, baseball) * $\log_2$[p(computer, baseball) / (p(computer) * p(baseball))]

$= (N_{AB}/N) * \log_2[(N_{AB}/N) / (((N_A - N_{AB}) / N) * ((N_B - N_{AB}) / N))]$

$= (23 / 26394) * \log_2[(23/26394) / (((1390 - 23) / 26394) * ((2144 - 23) / 26394))]$

$= (0.00087) * \log_2[(0.00087) / ((0.0518) * (0.0804))] = 0.00087 * \log_2(0.2089) = -0.002$

c) Given that the words "computer" and "program" appear in the data together 349 times and "computer" and "baseball" appear 23 times, it makes sense that the probability of "computer" and "baseball" would be lower than the probability of "computer" and "program".

d) The results of the data set are:

1. (('january', 'paper'), 181)
2. (('language', 'programming'), 153)
3. (('january', 'time'), 150)
4. (('january', 'program'), 149)
5. (('january', 'systems'), 149)
6. (('data', 'january'), 142)
7. (('january', 'presented'), 141)
8. (('january', 'programming'), 139)
9. (('program', 'programs'), 133)
10. (('january', 'method'), 125)

e) The results of the top 5 words that have the highest mutual info (with programming):

1. ((language, programming), 0.10326)
2. ((programming, languages), 0.09290)
3. ((program, programming), 0.06444)
4. ((programming, paper), 0.05984)
5. ((programming, programs), 0.05273)

These results appear to be reasonable given all the data from the collection.

6. Kullback-Leibler Divergence

a) 1.) KL-Divergence will be 0 when the two distributions are the same, so the KL-Divergence is always non-negative value, thus making the range between 0 and infinity.
2.) The KL-Divergence is equal to 0 only when p = q.

b) KL divergence measures the "distance" between two distributions, meaning they are not symmetric because D(P, Q) – D(Q, P) would have to be greater than 0, thus making D(p||q) =/= D(q||p).

c) If an event has q(x) = 0, then log(p(x) / q(x)) doesn't work since we would be dividing by 0. Since the KL-Divergence is the distance between distributions, and given q(x) = 0, then there is no distribution that can be made for q(x), meaning there is no distance between the distributions. This makes just the distribution of p(x) the KL-Divergence.