Bayes' Nets Written Questions:

A. 1.P(B|I,M) is not equivalent to P(I)P(B)P(M) because I is dependent on P(B) and P(M). Instead, we can assert P(B,I,M) is equivalent to P(B)P(M)P(I|B,M).

2.P(J|G) = P(J|G,I) is not implied by the above Bayes' net. We have:

P(J|G) = P(J|G,I)+P(J|G,¬I) since I and ¬I are disjoint events.

Now, for contradiction suppose P(J|G) = P(J|G,I). Then we would have:

P(J|G,I) = P(J|G,I) + P(J|G,¬I)

->0 = P(J|G,¬I) (Equation 1)

By the Bayes' net assumptions, and reading from the tables, we have:

P(J|G,¬I) = P(J,G,¬I) / P(G,¬I) by definition of conditional probability

= P(J|G) * P(G|¬I) * P(¬I) / P(G|¬I) * P(¬I)

= P(J|G)

= .9

Therefore, we have Equation 1 = 0 = .9, a contradiction -><-

The premise was contradictory, so then we must have P(J|G) != P(J|G,I)

3. P(M|G,B,I) = P(M|G,B,I,J) is implied by the given Bayes' Net. Starting from the right hand side, the definition of conditional probability gives:

P(M|G,B,I,J) = P(M,G,B,I,J) / P(G,B,I,J)

= P(M) * P(G|I,B,M) * P(B) * P(J|G) * P(I | B,M) / P(G|I,B) * P(B) * P(I|B)

*P(J | G) by the Bayes' Net assumptions

= P(M) * P(G|I,B,M) * P(B) * P(I | B,M) / P(G|I,B) * P(B) * P(I|B)

= P(M,G,B,I) / P(G,B,I) by Bayes' net assumptions

= P(M|G,B,I) by definition of conditional probability

Therefore, P(M|G,B,I) = P(M|G,B,I,J)

B. P(B,I,¬M,G,J)=P(B)P(!M)P(I|B,¬M)P(G|B,I,¬M)P(J|G)=.9*.9*.5*.8*.9=0.2916

C. First, we need to collapse I into G. We do this by lining up each value of the P(I|B,M) table with the two rows in the P(G|B,I,M) table that have matching B and M values. Then we create a table of P(G|B,M) values where each P(G|B,M)=P(G|B,I,M)P(I|B,M)+P(G|B,¬I,M)P(¬I|B,M). We now repeat this process to collapse G into J, creating a table of P(J|B,M)=P(J|G)P(G|B,M)+P(J|¬G)P(¬G|B,M). It is from this final table that we pluck the solution.

P(J|B,¬M): we know from the assumption that B=t and M=f and the table of P(I|B,M) that P(I=t,B=t,M=f)=.5 and likewise P(I=f,B=t,M=f)=.5. Note that as we collapse I into G, we only care about the rows in the P(G) table that have B=t and M=f since we are still assuming B=t and M=f for each step. We find P(G|B=t,M=f) by marginalizing I, which is done by adding P(G|I=t,B=t,M=f)*P(I=t,B=t,M=f)+P(G|I=f,B=t,M=f)*P(I=f,B=t,M=f)=.8*.5+.0*.5=.4. We can then quickly conclude that P(G|B=t,M=f)=.6. We can now find P(J|B=t,M=f)=P(J|G=t)P(G=t|B=t,M=f)+P(J|G=f)P(G=f|B=t,M=f)=.9*.4+.0*.6=.36.

D. We propose that Likelihood Weighting would be the best sampling method in this case. Since we have conditioned on guilty verdicts only, we will only sample from that subset. Also, since we

take evidence into account as we generate samples, we do not waste time generating samples that would not help us.

Naive Bayes Spam Classification:

We implemented the Naive Bayes filter by creating a “Classifier” class which trains for either spam or ham by taking in all of the samples from the corresponding training set, parsing each sample into a sequence of characters delimited by blank space, and hashing those sequences of characters as keys into a HashMap whose values indicate the number of occurrences of the sequence across the training documents of the relevant class.

After making 2 Classifiers spam and ham, we sum the total word count observed by each Classifier to identify the total number of words contained in the training collection.

We then create a lexicon by merging the HashMap of both Classifiers, and filtering out all words that occur k or fewer times.

Each Classifier then uses the lexicon to construct yet another HashMap with the same keys whose values are now the sum of the number of occurrences of that word in the relevant class (spam or ham) plus an input “m” value divided by the sum of the total word count of training documents of the relevant class and the product of the number of words in the lexicon with m. This results in a HashMap that indicates the Laplacian-smoothed probability of a word occurring in an email given that the email belongs to the Classifier’s class. Let us call this final HashMap the “probability map”.

In order to actually classify an email as being spam or ham, we hand it to the corresponding Classifier object and parse it just as we did in the training by separating it into sequences of characters delimited by blank space. The true probability that the email belongs to the Classifier’s class is replaced by summing the logs of the following (because, as we discussed in class, underflow could easily occur and break our classification if we use the raw product of probabilities):

1. The probability that the email is of that class statistically (which is .5, as the training collection is evenly distributed with spam and ham)
2. For every parsed token that occurred in an email of the given class, its probability map value
3. For every parsed token that didn’t occur in an email of the given class, m/(m*V+t), where m is an input variable for Laplacian Smoothing, V is the number of words in our lexicon, and t is the total word count of training documents of the relevant class

The Classifier who returns the higher sum of such logs logically has the higher true probability of the email belonging to its class.

We experimented with every combination of k and m and our best results resulted from k=8 and m=45. This gave a spam accuracy of 79% and a ham accuracy of 98% (output for the run is reproduced below) for an overall accuracy of (79+98)/2 = 88.5%. Also, it was noticed that there are 100 spam training documents and 100 ham training documents. This implied that P(spam) = P(ham) = .5. Because of this very specific case, it is reasonable to assert that ML classification will have no effect over MAP classification. ML classification does not give any weight to P(class), it simply takes the most likely class for a word given that word, i.e. P(word|class). In

MAP, we weight each conditional probability by the probability of that class. However, since P(spam) = P(ham), performing this weighting does nothing to change the relative probabilities of each word: the most probable spam/ham words will remain the most probable spam/ham words. Note that this is the **only** case where this can occur given our two class structure, as since P(ham) = 1-P(spam) for any P(ham) != .5 we must have P(ham) != P(spam) and the same reasoning holds if P(spam) != .5

OUTPUT:

SPAM RESULTS
percent of contents classified as spam: 79
percent of contents classified as ham: 21
HAM RESULTS
percent of contents classified as spam: 2
percent of contents classified as ham: 98

Two spam emails from testing that our model incorrectly classified as ham were spamtesting/0130.2004-01-01.GP.spam.txt and spamtesting/2658.2004-10-28.GP.spam.txt. The former was merely

"Subject: re : feeling tired ?",

which makes sense, because there is very little information to glean from a mere 4 words and 2 pieces of punctuation, and in addition none of the contents are particularly iconic of spam. The latter was

"Subject: important auction info
r 3 move
market research 8721 santa monica
boulevard # 1105 los angeles , ca 90069 - 4507",

which, although clearly spam to a human, could reasonably confuse the model seeing as its contents are not exactly typical of general spam: it involves a specific place and has serious words like "research" and "info", which are more indicative of ham and could reasonably fool our model.

Two spam emails that our model correctly classified were spamtesting/0195.2004-01-12.GP.spam.txt and spamtesting/0449.2004-02-14.GP.spam.txt. The former was

"Subject: . inc ; rease * ; * d . ic - k l * engt - h kslkbkcrbavoc
loading please wait . . . . . do you want a longer
penis ?
enlarge your penis ! instant rock hard
erections ! longer lasting time !
click here for information
remove
me",

which is quite clearly spam and abundant with keywords you would see in spam but never ham, so it follows that our model would place a higher likelihood of the email being spam. The latter was

“Subject: catv meaty cretin whirligig lorelei baptismal oscillate beaver brenda staunton heresy maroon rio coventry sexual crummy transalpine acme loose prison cranky cobra guile edgy stephanie premature tutu cia satire alia animadversion hire simplicial auric digression sunbonnet
hurtle angstrom cap probabilist sanicle doublet industrial bootlegger illusive embarrass reek neuroanatomy suspend cartesian atlantic handel
adobe photoshop cs 8 . 0 - $ 80
quark express 6 . 0 - $ 60
norton antivirus professional 2004 - $ 15
more titles available
software on the net
grab your copy
no
schroedinger donate stack corralled bowman compulsory chock catcall physiotherapist neglecter pacifism embark slaughterhouse”,

which, although rather nonsensical, contains some key features that our model should recognize as spam: the dollar sign, mentions of software (which seem to be somewhat prevalent across the spam training documents), and words like “more”, “available”, and “grab”.

Our reasoning for the inaccuracy of ham classification falls in the same vein. We suspect that the main cause was the inclusion of patterns prevalent in spam. For example, hamtesting/2151.2000-09-05.farmer.ham.txt was incorrectly classified as spam and contained a bizarre number of question marks and a significant repetition of the word “neon”, both of which could be feasibly attributed to spam:

“Subject: neon september 13
hey guys and girls ,
?
here ' s an idea for an icebreaker activity for our 2 nd week of neon . ? this is only an idea , you can use it or do something totally different . ? again , let me know along the way if you have something that has worked well for your group so we can share in the wealth .
?
someone in your group should have a couple of classic books full of icebreaker - game type of activities , building community in youth groups , and youth group trust builders . ? we all at one time had a selection of icebreakers in a gray binder that you might still have somewhere in a drawer . ? spend time in the games and forced interactions , especially over the beginning of the semester . ? if you don ' t have these resources , let me know . ? even if some of these ideas may seem familiar to you , remember that you may have been doing this far longer than the oldest kids have been in your group . ? but again , if you have any ideas for resources let me know .
?
in addition to the ideas to think about at the top of the 2 nd weeks topic , a

couple of additional thoughts given me by leaders :
? ? ? - realize and possibly communicate in a low key manner to the freshman that their expectations of neon may not initially be met . ? they may be coming in with ideas of what neon is , only to possibly be disappointed . ? in a sense it is similar to going to a restaurant or movie after tons of people have recommended it to you , and being a little let down . ? more importantly , they need to realize that neon is not an event , it is a process . ? the teens who say they love it are most often the ones who have been in it long enough to have built relationships and to have shared a bunch of common experiences . ? with time and a commitment on their part to be at neon weekly , their expectations will ultimately be met .
? ? ? - remember that the best quality time you have with the kids is probably before or after neon , or at other non - neon times .
?
have fun loving on your kids wednesday !
?
bobby
- neon opening lesson . doc”

Then there was hamtesting/0637.2000-03-20.farmer.ham.txt:

“Subject: hey ! guess it was hard to get back , huh ! i was going to get jake one of those baby wipe warmers and send it with art & jo , but walmart was out of them . maybe i need to get something else . do you have one of those toys that goes on the bar of the carrier ? is jake ok ? let me hear from you , please . i ' m making lacy stay off the phone from 7 - 9 : 30 every night so that you & shane can have a chance to call if you want to . i sure do miss holding jake and playing with cole . cole is so funny . i ' ve been telling everybody things that he does & they can ' t believe it at 2 yrs . old . i gotta get busy - - let me know if you get this . i know you ' re busy , just a little answer to let me know ' cause i sent you one a while back that you never answered and the last 2 that i sent shane & shelly they didn ' t get .
love ya , ma
get your private , free email at http : / / www . hotmail . com”

This one is a little tough to justify, but there are certain features that could crop up in spam relatively often, like “gotta”, “hey”, “have”, “busy”, and “answered”, things that suggest an intent to distract and a desire for attention and response.

Our ham classification was fine on emails such as hamtesting/4734.2001-07-09.farmer.ham.txt:

“Subject: frontrea plant , gas
when : wednesday , july 11 , 2001 2 : 00 pm - 3 : 00 pm (gmt - 06 : 00) central time (us & canada) .
where : 3127
* ~ * ~ * ~ * ~ * ~ * ~ * ~ * ~ * ~ *”

and hamtesting/4861.2001-09-05.farmer.ham.txt:

“Subject: kaf storage project

as requested , this is the summary of what we have discussed regarding kaf deal .
scenario 1 :
hpl nominates 6 , 200 mmbtu / day , apr ' 02 - may ' 02 , (409 , 000 mmbtu total) of their south texas production into the kathleen anne field in wilson county , tx . and enron replaces this same volume in the location of their choice .
scenario 2 :
scenario 1 plus ena sells hpl 500 , 000 mmbtu of storage service with an injection rate of approximately 15 , 000 mmbtu / day and a withdraw rate of 37 , 500 mmbtu / day to help them serve the swing on the tuffco contract .
notes : the field is not yet connected to hpl , but will be .
thanks for your help .
george huan
gas structuring“,

which are arguably indicative of ham from our model's perspective because they contain words and patterns that one would expect from colloquial dialogue and work-related discussion: observe the dates and times mentioned throughout, specific numbers, and mentions of serious companies and business-related acronyms like “mmbtu”, for example. Ham emails are clearly directed at individuals, or specific groups, to whom content will be personally engineered. Spam, meanwhile, relies on a smaller pool of words that are used to appeal to a large, general crowd of recipients.

Extra Credit

1. In addition to the above, we implemented a command line option to allow the user to use n-grams as the smallest unit of comparison for our classification algorithm. Selecting 1-grams results in exactly the same behaviour as before (given k and m values are the same), since a 1-gram is simply a word. For 2-grams and higher, we observed a tradeoff between increasing Ham accuracy and decreasing Spam accuracy. Our reasoning for why this is so is that Ham usually will be a logical sequence of words while Spam is often a garbled mess of random words. Higher n-grams mean fewer total grams to work with. If we assume that approximately, for all n, grams in spam are independent from each other (very loose assumption!) then having fewer grams for Spam training means less accuracy for Spam. In contrast, if we loosely assume that, because of common sentence structures, words etc., grams in Ham are dependent on each other, then for some large N and some intermediate n for $1<n<N$ we will achieve better accuracy by considering n-grams instead of 1-grams because the dependencies between words will be more evident.

Individual Contributions

Max: I implemented n-grams and did parsing for both our original assignment and n-grams. Also helped with number 1. Jar/ Readme creation/deployment

Fred: I worked on part 1, pair-programmed part 2, wrote a fair bit of this report, and helped debug part 2 ad nauseam.

Tyler: I helped with part 1 and wrote out a few of the proofs. Pair Programmed the non-extra credit, non-parsing portion of part 2 with Fred.