Question 1(8 points):
Consider again the example application of Bayes rule in Section 6.2.1 of Tom Mitchell's textbook. Suppose the doctor decides to order a second laboratory test for the same patient and suppose the second test returns a positive result as well. What are the posterior probabilities of cancer and -cancer respectively following these two tests? Assume that the two tests are independent.

Answer:

1. Finding P(++|cancer) and P(+ + |-cancer):
   a. P(+ + | cancer) = P(+|cancer) x P(+|cancer) = 0.98 x 0.98 = 0.9604.
   b. P(+ + | -cancer) = P(+ + | -cancer) x P(+ + | -cancer) = .03 x .03 = .0009.
2. Use law of total probability to find P(+ +):
   a. P(+ +) = P(+ + | cancer)P(cancer) + P(+ + | -cancer) P(-cancer) =
   b. .9604 x .008 + 0.0009 x .992 = 0.007683 + 0.00893 = 0.008576
3. Applying Bayes Theorem to P(cancer | + +):
   a. P(cancer | + +) = (P(+ + | cancer) P(cancer))/P(+ +) = (.9604 x .008) / .008576 = 0.8958955
   b. P(-cancer | ++) = (P(+ + | -cancer)P(-cancer)) / P(++) = (.0009 * .992) / .008576 = 0.1041045
4. **Therefore, the posterior probability of P(cancer) = 89.6% and the posterior probability of P(-cancer) = 10.4%**

Question 2(8 points):
Section 6.9.1 of Tom Mitchell's textbook demonstrates an example using the Naive Bayes Algorithm to predict a new instance based on a dataset with 14 examples from Table 3.2 of Chapter 3 of the book. If we only have 12 examples as shown below, what is the prediction results for the same new instance? Show your calculation.

New instance: <Outlook=sun, Temperature=cool, Humidity=high, Wind=strong>

Day	*Outlook*	*Temperature*	*Humidity*	*Wind*	*PlayTennis*
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes

Answer Below:

1. Get current probabilities:
   a. P(Yes) = 8/12 = 0.67
   b. P(No) = 4/12 = 0.33
   c. P(Sun | Yes) = 2/8 = 0.25
   d. P(Sun | No) = ¾ = 0.75
   e. P(Cool | Yes) = ⅜ = 0.375
   f. P(Cool | No) = ¼ = 0.25
   g. P(High | Yes) = ⅜ = 0.375
   h. P(High | No) = ¾ = 0.75
   i. P(Strong | Yes) = ⅜ = 0.375
   j. P(Strong | No) = 2/4 = 0.5

2. Get posterior probabilities:
   a. Posterior P (Yes) = P(Yes) x P(Sun | Yes) x P (Cool | Yes) x P(High | Yes) x P(Strong | Yes) = 0.67 x 0.25 x 0.375 x 0.375 x 0.375 = 0.0088
   b. Posterior P(No) = P(No) x P(Sun | No) x P(Cool | No) x P(High | No) x P(Strong | No) = 0.33 x .75 x .25 x .75 x .5 = 0.023

3. Normalize probabilities:
   a. P(yes) = .0088 / (.0088 + .023) = .0088 / .0318 = **27.7%**
   b. P(No) = .023 / (.0088 + .023) = .023 / .0318 = **72.3%**
4. **Therefore, the predicted result is “No.”**

<u>Question 3 (14 Points):</u> Answer question 4.7(page 125) of Tom Mitchell’s textbook as quoted below:

Consider a two-layered feedforward ANN with two inputs a and b, one hidden unit c, and one output unit d. This network has five weights (wca, wcb, wc0, wdc, wd0), where wx0 represents the threshold weight for unit x. Initialize these weights to the values (.1, .1, .1, .1, .1), then give their values after each of the first two training iterations of the BACKPROPAGATION algorithm. Assume learning rate n=.3, momentum A = 0.9, incremental weight updates, and the following training examples:

A b d
1 0 1
0 1 0

Iteration 1
Calculate inputs:
1. 0c = Wc0 + wca*a + wcb*b = 0.1 + 0.1 + 0 = 0.2
2. C = 1/(1 + e^-0.2) = 1/1.8187 = 0.5498
3. 0d = (wd0 + wdc * c) = 0.1 + 0.1 * 0.5498 = 0.15498
4. 1/(1+e^-0.15498) = 1 / (1 + 0.8564) = 0.5387

Calculate hidden layer errors

1. Errord = 0.5387 * (1 - 0.5387)^2 = .5387 * .4163^2 = 0.11464
2. Errorc = .5498 * (1 - .54983) * (0.1 * 0.11464) = .5498 * .4502 * .011464 = .00284

Calculate error for each hidden unit:

1. ∆wca = n * errorc * a + A * ∆wca(1-1) = 0.2 * .00284 * 1 + 0.9 * 0 = .2 * .00284 = .00057
2. ∆wcb = n * errorc *b + A * ∆wcb(1-1) = 0.2 * 0.00284 * 0 + 0.9 * 0 = 0
3. ∆wc0 = n * errorc * a + A * ∆wc0(1-1) = 0.2 * .00284 * 1 + 0.9 * 0 = .2 * .00284 = .00057
4. ∆wdc = n * errord * a + A * ∆wdc(1-1) = 0.2 * 0.11464 * 0.5498 + 0.9 * 0 = .2 * .11464 * .5498 = 0.0128
5. ∆wd0 = n * errord * x0 + A * ∆wd0(1-1) = 0.2 * 0.11464 * 1 = 0.02293

Update network weights:

1. Wca = wca + ∆wca = 0.1 + 0.00057 = 0.10057
2. Wcb = wcb + ∆wcb = 0.1 + 0 = 0.1
3. Wc0 = wc0 + ∆wc0 = 0.1 + 0.00057 = 0.10057
4. Wdc = wdc + ∆wdc = 0.1 + 0.01261 = 0.1128
5. Wd0 = ∆wd0 + Wd0 = 0.1 + 0.02293 = 0.12293

Iteration 2:
Calculate inputs:

5. 0c = Wc0 + wca*a + wcb*b = 0.10057 + 0.10057 * 0 + 0.1 * 1 = 0.20057
6. 1/(1 + e^-0.20057) = 1/1.8183 = 0.54997
7. 0d = (wd0 + wdc * c) = 0.12293 + 0.11261 * 0.5498 = 0.1848
8. 1/(1+e^-0.1848) = 1 / (1 + 0.831) = 0.546

Calculate hidden layer errors

3. Errord = 0.546 * (1 - 0.546)^2 = .546 * .45392^2 = 0.11254
4. Errorc = .54997 * (1 - .54997) * (0.1128 * 0.11254) = .45002 * .11261 * .011252 = .003

Calculate error for each hidden unit:

6. ∆wca = n * errorc * a + A * ∆wca(2-1) = 0.2 * .003 * 0 + 0.9 * .003 = .0027
7. ∆wcb = n * errorc * b + A * ∆wcb(2-1) = 0.2 * .003 * 1 + 0.9 * 0 = 0.0006
8. ∆wc0 = n * errorc * b + A * ∆wc0(2-1) = 0.2 * .003 * 1 + 0.9 * 0.003 * 1 = .2 * .003 + .0027 = .0033
9. ∆wdc = n * errord * a + A * ∆wdc(2-1) = 0.2 * 0.1128 * 0.546 + 0.9 * 0 = .2 * .1128 * .546 = .0123
10. ∆wd0 = n * errord * x0 + A * ∆wd0(2-1) = 0.2 * 0 + 0.9 * 0.02293 * 1= .9 * .02293 = .0207

Update network weights:

6. Wca = wca + ∆wca = 0.10057 + .0027 = **0.10327**
7. Wcb = wcb + ∆wcb = 0.1 + .0006 = **0.1006**
8. Wc0 = wc0 + ∆wc0 = 0.10057 + 0.0033 = **0.10387**
9. Wdc = wdc + ∆wdc = 0.11261 + .0123 = **0.12491**
10. Wd0 = ∆wd0 + Wd0 = 0.12293 + 0.02067 = **0.1436**