SEng 474 / CSc 578D

Data Mining – Spring 2018

Assignment 1 - Solutions

1. a) Construct the root and the first level of a decision tree for the contact lenses data using the ID3 algorithm.

For the root we have the following choices: Age

young: 2/4/2 (8) pre-presbyopic: 1/5/2 (8) presbyopic: 1/6/1 (8)

Spectacle-Prescription myope: 3/7/2 (12)

hypermetrope: 1/8/3 (12)

Astigmatism yes: 4/8/0 (12)

no: 7/5/0 (12)

Tear-prod-rate normal: 4/3/5 (12)

reduced: 0/0/12 (12)

x/y/z means that we have x instances of some class, y instances of another class, and z instances of yet another class. The order x/y/z doesn't matter for computing entropies. **NOTE:** Entropy is calculated with log base 2 ($\log_2$).

Age entropies: entropy(2/4/2) = $(-(2/8)*\log_2(2/8)-(4/8)* \log_2 (4/8)-(2/8)* \log_2 (2/8))$ = 1.5 entropy(1/5/2) = $(-(1/8)* \log_2 (1/8)-(5/8)* \log_2 (5/8)-(2/8)* \log_2 (2/8))$= 1.299 entropy(1/6/1) = $(-(1/8)* \log_2 (1/8)-(6/8)* \log_2 (6/8)-(1/8)* \log_2 (1/8))$= 1.061 avg_entropy = (8/24)*1.5 + (8/24)*1.3 + (8/24)*1.06 = 1.287 bits

Spectacle-Prescription entropies: entropy(3/7/2) = $(-(3/12)* \log_2 (3/12)-(7/12)* \log_2 (7/12)-(2/12)* \log_2 (2/12))$= 1.384 entropy(1/8/3) = $(-(1/12)* \log_2 (1/12)-(8/12)* \log_2 (8/12)-(3/12)* \log_2 (3/12))$= 1.1887 avg_entropy = (12/24)* 1.384 + (12/24)* 1.1887 = 1.28635 bits

Astigmatism entropies: entropy(4/8/0) = $(-(4/12)* \log_2 (4/12)-(8/12)* \log_2 (8/12)-0)$= .918 entropy(7/5/0) = $(-(7/12)* \log_2 (7/12)-(5/12)* \log_2 (5/12)-0)$=

.9799 avg_entropy = (12/24)* .918 + (12/24)* .9799 = .94895 bits

Tear-prod-rate entropies: entropy(4/3/5) = (-(4/12)* $\log_2$ (4/12)-(3/12)* $\log_2$ (3/12)-(5/12)* $\log_2$ (5/12))= 1.555 entropy(0/0/12) = (0-0-0)= 0 avg_entropy = (12/24)* 1.555 + (12/24)*0 = **.7775 bits**

The smallest average entropy is for Tear-prod-rate, so we choose it for the root.

Now we have two branches (Tear-prod-rate=reduced) and (Tear-prod-rate=normal). The first branch is actually a leaf because all of the instances going to that branch are "contact-lenses = none". For the second branch (Tear-prod-rate=normal) we have the following data instances:

age

pre-presbyopic presbyopic young young pre-presbyopic presbyopic presbyopic pre-presbyopic pre-presbyopic presbyopic young

young

spectacle- prescrip myope yes myope yes hypermetrope yes myope yes hypermetrope yes hypermetrope yes myope no hypermetrope no myope no hypermetrope no hypermetrope no myope no

We have the following choices to split further: Age

young: 2/2/0 (4) pre-presbyopic: 1/1/2 (4) presbyopic: 1/2/1 (4)

Spectacle-Prescription myope: 1/2/3 (6)

hypermetrope: 3/1/2 (6)

Astigmatism yes: 4/2/0 (6)

no: 1/5/0 (6)

Age entropies: entropy(2/2/0) = (-(2/4)* $\log_2$ (2/4)-(2/4)* $\log_2$ (2/4)-0)= .999 entropy(1/1/2) = (-(1/4)* $\log_2$ (1/4)-(1/4)* $\log_2$ (1/4)-(2/4)* $\log_2$ (2/4))= 1.5 entropy(1/2/1) = (-(1/4)* $\log_2$ (1/4)-(2/4)* $\log_2$ (2/4)-(1/4)* $\log_2$ (1/4))= 1.5 avg_entropy = (4/12)* .999 + (4/12)* 1.5 + (4/12)* .1.5 = 1.333 bits

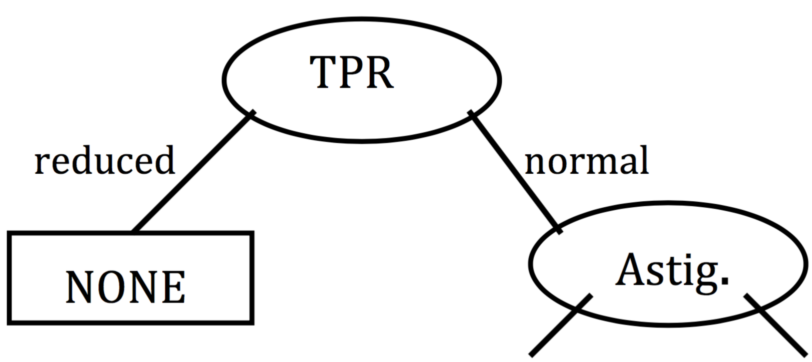
astigmatism tear-prod-rate contact-lenses

normal hard normal hard normal hard normal hard normal none normal none normal none normal soft normal soft normal soft normal soft normal soft

Spectacle-Prescription entropies: entropy(1/2/3) = (-(1/6)* $\log_2$ (1/6)-(2/6)* $\log_2$ (2/6)-(3/6)* $\log_2$ (3/6))= 1.459 entropy(3/1/2) = (-(3/6)* $\log_2$ (3/6)-(1/6)* $\log_2$ (1/6)-(2/6)* $\log_2$ (2/6))= 1.459 avg_entropy = (6/12)* 1.459 + (6/12)* 1.459 = 1.459 bits

Astigmatism entropies: entropy(4/2/0) = (-(4/6)* $\log_2$ (4/6)-(2/6)*log2(2/6)-0)= .918 entropy(1/5/0) = (-(1/6)* $\log_2$ (1/6)-(5/6)* $\log_2$ (5/6)-0)= .65 avg_entropy = (12/24)* .918 + (12/24)*.65 = .784 bits

So, we choose astigmatism for the next level node. The tree so far is:



b) There were multiple things we were hoping to see. Here is a list of the most important:

1) sklearn library does not use ID3 to build its decision tree but an algorithm named

CART (classification and regression tree).

2) CART builts binary trees, which imply that classes and attributes having more than two possible values will be evaluated under a series of combinations. For our data, it evaluates “none” vs. “soft+hard”, “soft” vs. “none+hard” and “hard” vs. ”none+soft”; and similarly on attributes who have more than two values. For instance the root calculation is: entropy(20/4) = (-(20/24)* $\log_2$ (20/24)-(4/24)*$\log_2$(4/24))= 0.65 entropy(9/15) = (-(9/24)* $\log_2$ (9/24)-(15/24)*$\log_2$(15/24))= 0.9544 entropy(19/5) = (-(19/24)* $\log_2$ (19/24)-(5/24)*$\log_2$(5/24))= 0.7383 avg_entropy = (0.65 + 0.9544 + 0.7383)/3 = 0.7809 bits

3) The way our data was encoded transformed our attributes into a larger set of binary attributes, and entropy was calculated on each.

Q2: Classifier Accuracy

a) Assume the data is equally split between the three classes (33.3% "red", 33.3% "blue" and 33.3% "yellow") and your classifier systematically predicts "red" for every test instances, what is the expected error rate of your classifier? (Show your work)

expected error : "red" = 0% of 33.3% = 0% misclassified
"blue" = 100% of 33.3% = 33.3% misclassified
"yellow" = 100% of 33.3% = 33.3% misclassified
total = 33.3% + 33.3% = 66.67% (or 2/3) misclassified

b) What if instead of always predicting "red", the classifier predicted "red" with a probability of 0.7, and "blue" with a probability of 0.3. What is the expected error rate of the classifier in this case?

expected error : "red" = 30% of 1/3 = 1/10
"blue" = 70% of 1/3 = 7/30
"yellow" = 100% of 1/3 = 1/3
total = 1/10 + 7/30 + 1/3 = 66.67% (or 2/3)

c) Now let's assume that the data is not split equally, but has half (1/2) of its data labeled "red", one fourth (1/4) labeled as "blue", and one-fourth (1/4) labeled as "yellow". What is the expected error rate of the classifier if, as in question a), the prediction is "red" for every test instances.

expected error : "red" = 0% of 1/2 = 0
"blue" = 100% of 1/4 = 1/4
"yellow" = 100% of 1/4 = 1/4
total = 0 + 1/4 + 1/4 = 50% (or 1/2)

d) With this dataset (half (1/2) labeled "red", one-fourth (1/4) labeled "blue", and one-fourth (1/4) labeled "yellow") What is the expected error rate of the classifier if, as in question b), it predicted "red" with a probability of 0.7, and "blue" with a probability of 0.3. expected error : "red" = 30% of 1/2 = 3/20 (or 0.15)
"blue" = 70% of 1/4 = 7/40 (or 0.175)
"yellow" = 100% of 1/4 = 1/4
total = 3/20 + 7/40 + 1/4 = 57.5 % (or 23/40)

Q3: MLE and MAP

These calculations follow exactly as we did in class

A) Theta = 7/10
B) beta2 = 4, Theta = 10/16
C) Formula for the mean is beta1/(beta1+beta2). Thus, Beta2 = 4 (again) and Theta = 10/16 again. Note that this will not always be the case.

Q4: Gradient Descent updates

A) Error function

$$E(X) = \frac{1}{2N} \sum_{i=1}^{N} (y_i - w_0 - w_1 x_{i,1} - w_2 x_{i,2}^4)^2$$

For all updates, w = w - \kappa* d/dw E(w)

B)

$$\frac{\partial}{\partial w_0} E(X) = -\frac{1}{N} \sum_{i=1}^{N} (y_i - w_0 - w_1 x_{i,1} - w_2 x_{i,2}^4)$$

C)

$$\frac{\partial}{\partial w_1} E(X) = -\frac{1}{N} \sum_{i=1}^{N} (y_i - w_0 - w_1 x_{i,1} - w_2 x_{i,2}^4) x_{i,1}$$

D)

$$\frac{\partial}{\partial w_2} E(X) = -\frac{1}{N} \sum_{i=1}^{N} (y_i - w_0 - w_1 x_{i,1} - w_2 x_{i,2}^4) x_{i,2}^4$$