Math 656 HW4

Jeff Gould

9/21/2020

1) This exercise will use the (nominal) weather data from WEKA. That data is also available on Canvas. You may check your answers using R or Weka, but you are expected to show the computations yourself, i.e. compute this by hand for Naïve Bayes. You need not do J48 calculations by hand.

a) What prediction would Naïve Bayes make for a day with: Outlook = sunny; Temperature = hot; Humidity = normal; windy = FALSE?

$P(play|sunny, hot, normal, windy') \approx P(sunny|play)P(hot|play)P(normal|play)P(windy'|play)P(play)$

```
weather <- foreign::read.arff("weather.nominal.arff")
play <- weather %>% filter(play == "yes")
Nsunny <- sum(play$outlook == "sunny")
Nhot <- sum(play$temperature == "hot")
Nhumid <- sum(play$humidity == "normal")
NnotWindy <- sum(play$windy == F)
```

$P(sunny|play)P(hot|play)P(normal|play)P(windy'|play)P(play) = (2/9)(2/9)(6/9)(6/9)(9/14) = \frac{8}{567}$

$P(play'|sunny, hot, normal, windy') \approx P(sunny|play')P(hot|play')P(normal|play')P(windy'|play')P(play')$

```
notPlay <- weather %>% filter(play == "no")
Nsunny <- sum(notPlay$outlook == "sunny")
Nhot <- sum(notPlay$temperature == "hot")
Nhumid <- sum(notPlay$humidity == "normal")
NnotWindy <- sum(notPlay$windy == F)
```

$P(sunny|play')P(hot|play')P(normal|play')P(windy'|play')P(play') = (3/5)(2/5)(1/5)(2/5)(5/14) = \frac{6}{875}$

$P(play|sunny, hot, normal, windy') = \frac{\frac{8}{567}}{\frac{8}{567}+\frac{6}{875}} = \frac{500}{743} \approx 0.6729$

b) b. (harder) Find a combination of the features for which Naïve Bayes gives a different answer than the J48 decision tree.

```
############ J48 ############
weatherJ48 = J48(play ~., data = weather)
weatherJ48
```

```
## J48 pruned tree
## ------------------
##
## outlook = overcast: yes (4.0)
## outlook = rainy
## |   windy = FALSE: yes (3.0)
## |   windy = TRUE: no (2.0)
## outlook = sunny
## |   humidity = high: no (3.0)
## |   humidity = normal: yes (2.0)
##
## Number of Leaves  :  5
```

```
##
## Size of the tree :   8
```

```r
#summary(weatherJ48)
```

Using a J48 decidion tree, we see that the prediction for conditions of $Outlook = rainy$ and $windy = TRUE$ is that $play$ will be yes

Let's use naive Bayes on those conditions, $Outlook = rainy$ and $windy = TRUE$, and set $humidity = high$ and $temp = hot$

$P(play|rainy, windy', humid, hot) = P(rainy|play)P(windy'|play)P(humid|play)P(hot|play)P(play)$

```r
play <- weather %>% filter(play == "yes")
Nrainy <- sum(play$outlook == "rainy")
Nhot <- sum(play$temperature == "hot")
Nhumid <- sum(play$humidity == "high")
Nwindy <- sum(play$windy == F)
```

$= (3/9)(6/9)(3/9)(2/9) = (1/3)(2/3)(1/3)(2/9)(9/14) = \frac{4}{378}$

$P(play'|rainy, windy', humid, hot) = P(rainy|play')P(windy'|play')P(humid|play')P(hot|play')P(play')$

```r
dontPlay <- weather %>% filter(play == "no")
Nrainy <- sum(dontPlay$outlook == "rainy")
Nhot <- sum(dontPlay$temperature == "hot")
Nhumid <- sum(dontPlay$humidity == "high")
Nwindy <- sum(dontPlay$windy == F)
```

$= (2/5)(2/5)(4/5)(2/5)(5/14) = \frac{32}{1750}$

$\frac{\frac{4}{378}}{\frac{4}{378}+\frac{32}{1750}} = 0.36656$

So under the conditions above, Naive Bayes would say there's only a 36.67% chance of playing, and classify it as $Don'tPlay$. However under the $J48$ rule, we would classify this instance as playing.

2) Cross-validation

One advantage of using synthetic data distributions for testing is that you can generate many points with the same underlying distribution. We will use that to explore cross-validation. Get the two data sets NF150A.arff and NF150B.arff from the Data folder on Canvas. Both are samples of 150 points from the Near-Far distribution that was used in class. Use J48 to classify the data in NF150A and predict the error rate both by testing on the training data and using 10-fold cross validation. Then measure the error rate by using NF150B as a test set. How well did the training data and cross validation predict the error rate on new data?

```r
NF150A <- foreign::read.arff("NF150A.arff")
NF150B <- foreign::read.arff("NF150B.arff")

NFA_J48 <- J48(Prox ~., data = NF150A)
summary(NFA_J48)
```

```
##
## === Summary ===
##
## Correctly Classified Instances         150              100      %
## Incorrectly Classified Instances         0                0      %
## Kappa statistic                          1
```

```
## Mean absolute error                      0
## Root mean squared error                  0
## Relative absolute error                  0      %
## Root relative squared error              0      %
## Total Number of Instances              150
##
## === Confusion Matrix ===
##
##    a   b   <-- classified as
##  126   0 |   a = Far
##    0  24 |   b = Near
```

```r
evaluate_Weka_classifier(NFA_J48, numFolds = 10)
```

```
## === 10 Fold Cross Validation ===
##
## === Summary ===
##
## Correctly Classified Instances         148               98.6667 %
## Incorrectly Classified Instances         2                1.3333 %
## Kappa statistic                          0.9504
## Mean absolute error                      0.0133
## Root mean squared error                  0.1155
## Relative absolute error                  4.8946 %
## Root relative squared error             31.4653 %
## Total Number of Instances              150
##
## === Confusion Matrix ===
##
##    a   b   <-- classified as
##  125   1 |   a = Far
##    1  23 |   b = Near
```

Using cross validation, we would expect to correctly classify 98.7% of instances correctly on out of sample data, with an error rate of 1.3%.

```r
NF150A$predictClass <- predict(NFA_J48, NF150A)

mean(NF150A$Prox == NF150A$predictClass)
```

```
## [1] 1
```

When running our model on the training data, we find that 100% of the instances were correctly classified. But we would expect a higher correct classification rate on the training data, and higher correct classification on the training data isn't always a good thing, as it ould be indicative of overfitting.

Next we try the model on the test data. Here we find that the model correctly classified 98% of the data points (147/150), or an error rate of 2%, which is very similar to our results from the cross-validation.

```r
NF150B$predictClass <- predict(NFA_J48, NF150B)

mean(NF150B$Prox == NF150B$predictClass)
```

```
## [1] 0.98
```

```r
sum(NF150B$Prox == NF150B$predictClass)
```

```
## [1] 147
```