Homework 3

Kelsey Neis

10/8/2021

Question 1

You are given a task to evaluate how well a new fire mapping algorithm works. The fire mapping algorithm is a Bayesian classifier which labels all the locations into two classes, burned and unburned. To evaluate the algorithm, two regions are tested. The confusion matrices of these two regions are given in Table 1 and Table 2.

a) Calculate the TPR, FPR, Precision, Recall, and F-measure for the "burned" class for both Dataset 1 and Dataset 2

TPR (true positive rate):

$$\begin{aligned} TPR_{D1} &= \frac{TP}{TP+FN} = \frac{30}{30+20} = .6 \\ TPR_{D2} &= \frac{TP}{TP+FN} = \frac{30}{30+20} = .6 \end{aligned} \tag{1}$$

FPR (false positive rate):

$$FPR_{D1} = \frac{FP}{FP+TN} = \frac{10}{10+40} = .2 FPR_{D2} = \frac{1000}{1000+4000} = .2 \tag{2}$$

Precision:

$$Precision_{D1} = \frac{TP}{TP+FP} = \frac{30}{30+10} = .75 Precision_{D2} = \frac{30}{30+1000} = .029 \tag{3}$$

F-measure:

$$F1_{D1} = \frac{2 \times TP}{2 \times TP + FP + FN} = \frac{60}{60+10+20} = .667 F1_{D2} = \frac{60}{60+1000+20} = .0555 \tag{4}$$

b) Is there a difference in the values of the metrics evaluated in part (a) for the two datasets? If so, what characteristics of the data sets (that are used to derive the above contingency tables) lead to the differences between the values of (TPR, FPR) and (Precision, Recall, F-measure) that you observe above.

Yes, although they have the same TPR(recall) and FPR, the precision of Dataset 1 is much higher and the F1 measure for the second data set is much lower. Since the second data set has way more cases where the actual class is unburned, that proportion, compared with the relatively small number of cases in the burned class, makes for a highly skewed data set. Since Precision and F1-measure are sensitive to skew, those results differ greatly between D1 and D2.

Question 2

Consider a data set with instances belonging to one of two classes - positive (+) and negative (-). A classifier was built using a training set consisting of an equal number of positive and negative instances. Among the training instances, the classifier has a recall m=50% on the positive class and a recall of n=90% on the negative class. The trained classifier is now tested on two data sets. Both have similar data characteristics as the training set. The first data set has 1000 positive and 1000 negative instances. The second data set has 100 positive and 1000 negative instances.

a) Draw the expected confusion matrix summarizing the expected classifier performance on the two data sets.

$$\begin{aligned} Recall_{positive} &= \frac{TP}{TP+FN} = .5 \\ TP &= .5(TP+FN) \\ TP &= FN \\ Recall_{negative} &= \frac{TN}{TN+FP} = .9 \\ TN &= .9(TN+FP) \\ TN &= 9FP \end{aligned} \tag{5}$$

	Algorithm Output = (+)	Algorithm Output = (-)
True Label = (+)	500	500
True Label = (-)	100	900

	Algorithm Output = (+)	Algorithm Output = (-)
True Label = (+)	50	50
True Label = (-)	100	900

b) What is the accuracy of the classifier on the training set? Compute the precision, TPR and FPR for the two test data sets using the confusion matrix from part A. Also report the accuracy of the classifier on both data sets.

$$\begin{aligned} Precision_{D1} &= \frac{TP}{TP+FP} = \frac{500}{500+100} = .8\bar{3} \\ Precision_{D2} &= \frac{TP}{TP+FP} = \frac{50}{50+100} = .\bar{3} \\ TPR_{D1} &= Recall_{positive} = .5 \\ TPR_{D2} &= Recall_{positive} = .5 \\ FPR_{D1} &= \frac{FP}{FP+TN} = \frac{100}{100+900} = .1 \\ FPR_{D2} &= \frac{FP}{FP+TN} = \frac{100}{100+900} = .1 \\ Accuracy_{D1} &= \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} = \frac{1400}{2000} = .7 \\ Accuracy_{D2} &= \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} = \frac{950}{1100} = .8\bar{6}\bar{3} \end{aligned} \tag{6}$$

c) In the scenario where the class imbalance is pretty high, how are precision and recall better metrics in comparison to overall accuracy? What information does precision capture that recall doesn't?

Precision and recall can provide more information about accuracy with regard to class, as opposed to overall accuracy. This can be very important information to preserve, especially for applications such as testing for a disease.

Question 3

You are trying to evaluate four different COVID-19 tests, T1, T2, T3, and T4. These tests have been developed by different organizations, and their evaluations by these organizations have been reported in the following evaluation measures: TPR and FPR as follows

T1: TPR = 0.5 and FPR = 0.01, T2: TPR = 0.99 and FPR = 0.1, T3: TPR = 0.99 and FPR = 0.01, T4: TPR = 0.9 and FPR = 0.05.

Note that T1, T2 and T3 correspond to the classifiers T1, T2 and T3 introduced in the lecture note "chap4_imbalanced_classes.pdf" on pages 25-27

a) Calculate the number of True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN) using the classifier T4 for the following cases:

1. Balanced skew case in which we have 100 positives and 100 negatives (Note: Confusion matrices of T1, T2 and T3 for this case are provided on slide 25)

$$
\begin{aligned}
TPR_{T4} = \frac{TP}{TP+FN} &= .9 \\
TP &= 9FN \\
TP &= 90 \\
FN &= 10 \\
FPR_{T4} = \frac{FP}{FP+TN} &= .05 \\
.95FP &= .05TN \\
19FP &= TN \\
FP &= 5 \\
TN &= 95
\end{aligned}
\tag{7}
$$

	Algorithm Output = (Yes)	Algorithm Output = (No)
True Label = (Yes)	90	10
True Label = (No)	5	95

2. Medium skew case in which we have 100 positives and 1000 negatives (Note: Confusion matrices of T1, T2 and T3 for this case are provided on slide 26)

$$
\begin{aligned}
TP &= 9FN \\
TP &= 90 \\
FN &= 10 \\
19FP &= TN \\
FP &= 50 \\
TN &= 950
\end{aligned}
\tag{8}
$$

	Algorithm Output = (Yes)	Algorithm Output = (No)
True Label = (Yes)	90	10
True Label = (No)	50	950

3. High skew case in which we have 100 positives and 10000 negatives (Note: Confusion matrices of T1, T2 and T3 for this case are provided on slide 27)

$$
\begin{aligned}
TP &= 9FN \\
TP &= 90 \\
FN &= 10 \\
19FP &= TN \\
FP &= 500 \\
TN &= 9500
\end{aligned} \tag{9}
$$

	Algorithm Output = (Yes)	Algorithm Output = (No)
True Label = (Yes)	90	10
True Label = (No)	500	9500

b) Compute precision, recall, F-measure, TPR/FPR of T4 for each of the aforementioned three cases

Balanced:

$$
\begin{aligned}
Precision_1 &= \frac{TP}{TP+FP} = \frac{90}{90+5} = .947 \\
Recall_1 = TPR_1 &= \frac{TP}{TP+FN} = \frac{90}{90+10} = .9 \\
FPR_1 &= \frac{FP}{FP+TN} = \frac{5}{5+95} = .05 \\
F-measure_1 &= \frac{2\times TP}{2\times TP+FP+FN} = \frac{180}{180+5+10} = .923
\end{aligned} \tag{10}
$$

Medium skew:

$$
\begin{aligned}
Precision_2 &= \frac{TP}{TP+FP} = \frac{90}{90+50} = .643 \\
Recall_2 = TPR_2 &= \frac{TP}{TP+FN} = \frac{90}{90+10} = .9 \\
FPR_2 &= \frac{FP}{FP+TN} = \frac{50}{50+950} = .05 \\
F-measure_2 &= \frac{2\times TP}{2\times TP+FP+FN} = \frac{180}{180+50+10} = .75
\end{aligned} \tag{11}
$$

High skew:

$$
\begin{aligned}
Precision_3 &= \frac{TP}{TP+FP} = \frac{90}{90+500} = .153 \\
Recall_3 = TPR_3 &= \frac{TP}{TP+FN} = \frac{90}{90+10} = .9 \\
FPR_3 &= \frac{FP}{FP+TN} = \frac{500}{500+9500} = .05 \\
F-measure_3 &= \frac{2\times TP}{2\times TP+FP+FN} = \frac{180}{180+500+10} = .261
\end{aligned} \tag{12}
$$

c) Which classifier is strictly better than T4 (irrespective of skew)? Briefly explain why.

T3 has better F-measure and Precision across all levels of skew.

d) Which classifier is strictly worse than T4 (irrespective of skew)? Briefly explain why.

None of the other classifiers are strictly worse across all measures.

e) Which classifiers may be better or worse than T4 (depending on skew)? Briefly explain why.

T4 is better than T2 in medium and high skew cases, because the difference in the FP and TN labels is magnified by the skew. T1 is worse in low and medium skew (in most measures), but better in high skew. This is because the FP for T1 is a smaller proportion of the negative cases than T4, so that makes up for the poor TP to FN ratio that T1 has.

Question 4

Consider a data set with four binary attributes X1, X2, X3 and X4. The attribute X4 takes exactly the same value as X3 for each record, i.e., X4 is equal to X3. In each of the following three scenarios, find whether the decision boundary learnt by the two models would be similar, otherwise find which of the two models would perform better. Provide a brief justification for each.

(i) We build two Naïve Bayes models:

a) M1 that is learnt using all the four attributes.

b) M2 that is learnt using the three attributes X1, X2, and X3.

M2 will perform better than M1, because correlated attributes degrade the performance of Naïve Bayes models.

(ii) We build two ANN models:

a) M1 that is learnt using all the four attributes.

b) M2 that is learnt using the three attributes X1, X2, and X3.

M1 and M2 will perform similarly, since M1 doesn't have an excessive number of correlated attributes and redundant attributes receive the same weights.

Question 5

For each of the two given scenarios, make a right choice of K for the KNN classifier in order to obtain better performance with a brief explanation.

a) K = 1 or K=5 or K = 50?

K = 1 will be best. The decision boundary is large and clean enough where if a new data point is added and it's on one or the other side of the boundary, there's a good chance it's in the same class as its nearest neighbor. Furthermore, the plus class is more spread out, so if you were to increase K, there might be a risk that you'd capture many circles when looking for K nearest neighbors.

b) K = 1 or K = 5 or K=50?

K = 5 will ensure that the noise does not affect the choices made.

Question 6

a) If you had to choose between the Naïve Bayes and Decision Tree classifiers, which would you prefer for a classification problem where there are numerous missing values in the training and test data sets? Indicate your choice of classifier and briefly explain why the other one may not work so well?

I would choose Naive Bayes, since it not only handles missing values in the training set, but can more efficiently handle missing values in the test set by excluding the missing values from the computations, whereas the decision tree's methods for dealing with missing values leads to more splitting, and thus a more complex model.

b) Consider the problem of predicting whether a person is a good credit risk given the following attributes: hair color, income, weight, time in current job, marital status, height, age, and birth month. If you had to choose between Ripper and a k-nearest neighbor classifier, which would you prefer? Indicate your choice of classifier and briefly explain why the other one may not work so well?

I would use Ripper, because this would provide a hierarchical system in which some attributes are given more importance to making the decision. K-NN would simply not capture the greater significance of income than hair color when calculating distance between data points. Also, K-NN would not handle missing values as well as Ripper, and some of the attributes in this problem seem unlikely to have values for every data point.

Question 7

Given the data sets shown in Figure 2, explain how the decision tree, naïve Bayes (NB), and k-nearest neighbor (k-NN) classifiers would perform on these data sets.

1st figure: a decision tree would handle this data well, since the distinguishing attributes would be identified through the course of running the algorithm, so the first few splits would lead to relatively pure leaves. Naïve Bayes would handle the noise well, since it appears to be evenly distributed among Class A and Class B, which means that the noise attributes would not impact the posterior probability estimates. However it wouldn't work well overall, because the distinguishing attributes are not conditionally independent - if one of them is true, then we know that the rest are false. K-NN would not do well with the noisy attributes, since they would artificially change the proximity between instances. One way to deal with that could be to perform dimensionality reduction beforehand to get rid of the noisy attributes.

2nd figure: the decision tree would have difficulty with this data, because the two attributes are interacting: given the value of one attribute, this does not provide enough information to distinguish between classes. Naïve Bayes would also not handle this data well, since the two attributes are not completely conditionally independent. K-NN would be able to perform well, because K-NN handles interacting attributes well.