

# MEASURING PERFORMANCE IN SUPERVISED LEARNING 2

-APPLIED ANALYTICS-

APM Chapter 11.1 & 11.2, ISLR Chapter 2.2.3

Lecturer: Darren Homrighausen, PhD

# Preamble:

- Define what makes a good classifier
- Well-calibrated probabilities
- Confusion matrices
- Quantifying success in classification

# AN OVERVIEW OF CLASSIFICATION

Some examples:

- A person arrives at an emergency room with a set of symptoms that could be 1 of 3 possible conditions. Which one is it?
- A online banking service must be able to determine whether each transaction is fraudulent or not, using a customer's location, past transaction history, etc.
- Given a set of individuals sequenced DNA, can we determine whether various mutations are associated with different phenotypes?

All of these problems are **classification** problems.

# THE SET-UP

It begins just like regression: suppose we have observations

$$\mathcal{D} = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$$

Again, we want to estimate a model  $f$  that can take in a value  $X$  and predict as yet observed data

(In this context  $f$  is commonly called a **classifier**)

The same constraints apply:

- We want a classifier that predicts test data, not just the training data.
- Often, this comes with the introduction of some bias to get lower variance and better predictions.

# DEFINING PERFORMANCE FOR REGRESSION

Regression is defined as having  $Y$  that are quantitative, hence it makes sense to use squared error loss

$$\rightarrow \ell(f(X), Y) = (f(X) - Y)^2$$

We define  $f_*$ , which is the model we would construct if


- we had an infinite number of test observations
- and we could minimize the test error over all possible models



Note that we can exactly write down this model:

$$f_*(X) = \mathbb{E}[Y|X]$$

Any such  $f_*$  is called the “**Bayes’ rule** with respect to the loss”.

When the loss is squared error,  $f_*$  is called the **regression function** and is the conditional expectation of  $Y$  at  $X$  

# DEFINING NOTATION FOR CLASSIFICATION

Instead, if  $Y$  takes on a few **levels**, **labels**, or **classes**, then we are doing classification

**EXAMPLE:**  $Y$  takes on values  $C_1 = \text{'threat'}$  or  $C_2 = \text{'no threat'}$

The labels that  $Y$  takes on are arbitrarily named

Often, the labels are encoded as integers

**EXAMPLE:**  $Y$  takes on values  $-1$  or  $1$  or values  $0$  or  $1$

In general,  $Y$  can take on more than two levels. We will notate the total number as  $C$  and the levels as  $C_\ell$

# DEFINING PERFORMANCE FOR CLASSIFICATION

As  $Y$  takes only a few values, it makes sense to define a loss that answers the question: “Did my model make a mistake?”

This loss goes by the name **0-1 loss**:

$$\ell(\hat{Y}, Y) = \mathbf{1}(\hat{Y} \neq Y) = \begin{cases} 1 & \text{if } \hat{Y} \neq Y \\ 0 & \text{if } \hat{Y} = Y \end{cases}$$

**QUESTION:** What would happen if we used 0-1 loss for regression?

# DEFINING PERFORMANCE FOR CLASSIFICATION

The test error for regression looked like:

$$\text{test error} = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} (Y_i - \hat{Y}_i)^2$$

We will apply the same idea to 0-1 loss:

$$\text{test error} = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} \mathbf{1}(Y_i \neq \hat{Y}_i)$$

(For classification, this is also called the **misclassification rate**)

Note what happens when you average a bunch of 0-1 random numbers..



# DEFINING PERFORMANCE FOR CLASSIFICATION

```
> rbinom(10,1,.4)
[1] 1 1 0 1 0 0 1 0 0 1
> rbinom(10,1,.4)
[1] 0 0 0 0 0 0 1 1 1 1
> mean(rbinom(10,1,.4))
[1] 0.5
> mean(rbinom(10000,1,.4))
[1] 0.4001
```

Averaging a large number of 0-1s, you get the probability of getting a 1

In our case, this means that a small value of the loss  
→ small probability  $\hat{Y} \neq Y$

# DEFINING PERFORMANCE FOR CLASSIFICATION

In addition to the loss function, we can look at the proportion of times the test supervisor  $Y$  takes on particular values:

$$\frac{\#(Y = C_1)}{n_{test}}, \frac{\#(Y = C_2)}{n_{test}}, \dots, \frac{\#(Y = C_C)}{n_{test}}$$

again, we can look at what would happen as  $n_{test}$  got very large:

$$\frac{\#(Y = C_\ell)}{n_{test}} \rightarrow \text{prob}(Y = C_\ell) = p_\ell$$

A related concept in probability is called the **odds**:

$$\text{odds}_\ell = \frac{p_\ell}{1 - p_\ell}$$

(this definition of odds is the reciprocal of the 'lottery'-based definition)

# DEFINING PERFORMANCE FOR CLASSIFICATION

Many classifiers will seek to estimate  $p_\ell$ ,  $\hat{p}_\ell$ , and then turn these estimated probabilities into classifications

(Just like we wrote  $\hat{f}(X)$ , we will write  $\hat{p}_\ell(X)$  to be predicted probability at  $X$ )

The best possible classifier,  $Y_*$ , takes the label  $C_\ell$ , where  $p_\ell$  is the maximum probability

Hence, a good starting point is to apply the same rule to the estimated probabilities

**EXAMPLE:**  $Y$  takes on values  $C_1 = \text{'threat'}$  or  $C_2 = \text{'no threat'}$ .  
If  $p_1 > p_2$ , then  $p_1 > 1/2$ .  $\rightarrow Y_* = \text{'threat'}$

If we have probability estimates  $\hat{p}_1$  and  $\hat{p}_2$ , we can make an analogous rule:  $\hat{Y} = \text{'threat'}$  if  $\hat{p}_1 > \hat{p}_2$

# Well-calibrated probabilities

# WELL-CALIBRATED PROBABILITIES

Most classification methods report  $\hat{p}_\ell$

(Or can be transformed to have reported probabilities with e.g. the softmax function)

We might be interested only in the classifications themselves

(Example: Is there a threat in that image?)

Or, we might want the actual probabilities

(Example: Probability that someone clicks on an ad)

# WELL-CALIBRATED PROBABILITIES

Suppose we train a classification model on training data

Using a test set, we get the predicted probabilities

**SANITY CHECK:** Let's look at all the observations in the test set  $(X, Y)$  such that  $\hat{p}_\ell(X) = 0.2$

If we look at all the  $Y$ 's of those  $X$ 's, it's reasonable to expect that the proportion  $Y = C_\ell \approx \hat{p}_\ell(X)$

If this happens for all  $\ell$  and  $\hat{p}_\ell$ , we say that the  $\hat{p}_\ell$  are **well-calibrated**

# CALIBRATION PLOT

However, it's generally the case that there is at most 1 observation such that  $\hat{p}_\ell = 0.25$

(Or any value)

So, instead of asking for exact equality, we **bin** the probabilities

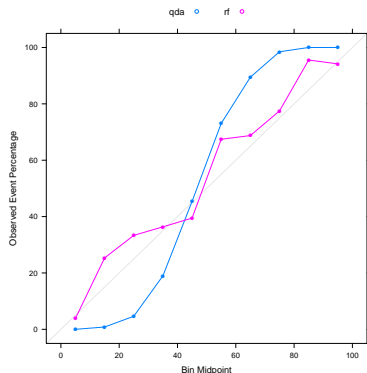
If we choose 10 bins, then we the calibration idea is:

Find all  $(X, Y)$  such that  $.1 \leq \hat{p}_\ell(X) < .2$ . Does the proportion of  $Y = C_\ell \approx 0.15$ ?

(Here, 0.15 is the midpoint)

# CALIBRATION PLOT

If  $C = 2$ , we can visualize this check with a **calibration plot**



```
testing$qda = predict(qdaFit, Xtest ...  
testing$rf  = predict(rfFit, Xtest ...  
calData1 = calibration(Ytest ~ qda + rf,  
                        data = testing, cuts = 10)
```



# EQUIVOCAL ZONES

Suppose we have  $C = 2$  and we get estimated probabilities  $\hat{p}_1 = 0.49$

We can make the prediction  $\hat{Y} = C_2$ , which would mimic  $Y_*$

However, we wouldn't be very confident in this prediction

We can make an informal confidence interval known as an **equivocal zone**:  $0.5 \pm z$ , where  $z$  could be 0.1

Now, when we go to turn the  $\hat{p}_\ell$  into classifications, we will skip any observation such that

$$0.5 - z \leq \hat{p}_1 \leq 0.5 + .1$$

(This can be extended to  $C > 2$  by making a similar interval around  $1/C$ )

# Displaying classifier results

## AN EXAMPLE

Suppose we are interested in predicting whether or not the economy will be in a **recession**

We have quarterly measurements of

- State level economic growth  
(Larger number is better)
- Federal level variables such as GDP, interest rates, employment, S&P 500, ...

Here, we will code the supervisor as

$$Y = \begin{cases} 1 & \text{if recession} \\ 0 & \text{if growth} \end{cases}$$

# CONFUSION MATRIX

We can report our results in a matrix:

		Truth		
		Recession	No Recession	Totals
Our Preds.	Recession	TP	FP	$P^* = TP + FP$
	No Recession	FN	TN	$N^* = FN + TN$
	Totals	$P = TP + FN$	$N = FP + TN$	$n_{\text{total}}$

The misclassification rate is

$$1 - \frac{TP + TN}{n_{\text{total}}} = \frac{FP + FN}{n_{\text{total}}}$$

# KAPPA SCORE

The **Kappa** score is the degree to which the classifications match the truth relative to what would be expected if they were independent

$$\kappa = \frac{O - E}{1 - E}$$

- $O = (TP + TN)/n_{\text{total}}$   
(This is 1 - misclassification rate)
- $E = ((TP + FP)/n_{\text{total}})((TP + FN)/n_{\text{total}})$   
(This strange formula is estimating the probability that the classifier and the truth would take the same level if they are independent)

Note that the Kappa score is only defined for  $C = 2$

(Also, it only makes sense if the confusion matrix is computed on data not used to train  $\hat{p}_\ell$ )

# SENSITIVITY AND SPECIFICITY

**SENSITIVITY:** The fraction of true positives (TP) out of the total number of actual positives (P)

(Notationally:  $TP/P$ )

**SPECIFICITY:** The fraction of true negatives (TN) out of the total number of actual negatives (N)

(Notationally:  $TN/N$ )

We can think of this in terms of hypothesis testing

$H_0$  : no recession

$H_A$  : recession

**SENSITIVITY:**  $\mathbb{P}(\text{reject } H_0 | H_0 \text{ is false})$  or  $1 - \mathbb{P}(\text{Type II error})$

(This is the same as power)

**SPECIFICITY:**  $\mathbb{P}(\text{accept } H_0 | H_0 \text{ is true})$  or  $1 - \mathbb{P}(\text{Type I error})$

# PRECISION AND RECALL

Other commonly used criteria are **precision** and **recall**

**PRECISION:** This is the fraction of true positives (TP) out of total number of predicted positives ( $P^*$ )

(Notationally:  $TP/P^*$ )

**RECALL:** This is the fraction of true positives (TP) out of the total number of actual positives ( $P$ )

(Notationally:  $TP/P$ . This is the same as sensitivity and power)

There is a combination of these two known as **F1 score**:

$$F1 = (2) \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

(This is the **harmonic mean** of precision and recall and  $0 \leq F1 \leq 1$ )

A larger F1 score indicates a **better** procedure

# Non-accuracy based criteria



# USES OF CLASSIFIERS

Usually, we want to make a decision with the output of classifiers

These decisions tend to have a cost/benefit analysis attached to them

**EXAMPLE:** Default rates. A credit card company wants to determine whether or not to issue credit

Perhaps the expected revenue from adding a new customer is \$1000 and the expected cost of a default is \$10000

# USES OF CLASSIFIERS

Let's consider "default" as the **positive**

Then the cost/benefit associated with various decisions are

		Truth	
		Default	No default
Our Preds.	Default	\$0	-\$1000
	No default	-\$9000	\$1000

		Truth		Totals
		Default	No default	
Our Preds.	Default	TP	FP	
	No default	FN	TN	
	Totals	$P = TP + FN$	$N = FP + TN$	

Then expected profit is

$$\text{Exp. Profit} = 0 \cdot p(TP) - 1000 \cdot p(FP) - 9000 \cdot p(FN) + 1000 \cdot p(TN)$$

Using a test set, we can estimate these quantities

$$\widehat{\text{Exp. Profit}} = 0 \cdot TP/P - 1000 \cdot FP/N - 9000 \cdot FN/P + 1000 \cdot TN/N$$

# USES OF CLASSIFIERS

		Truth		
		Default	No default	Totals
Our Preds.	Default	450	1200	
	No default	50	8300	
	Totals	500	9500	10000

$$\widehat{\text{Exp. Profit}} = 0 \cdot TP/P - 1000 \cdot FP/N - 9000 \cdot FN/P + 1000 \cdot TN/N$$

So, using our test results:

$$\begin{aligned}\widehat{\text{Exp. Profit}} &= -1000 \cdot 1200/9500 - 9000 \cdot 50/500 + 1000 \cdot 8300/9500 \\ &= -152.63\end{aligned}$$

→ We would have to do a lot better than that!

# Postamble:

- Define what makes a good classifier  
(Defining good depends on the purpose. It could have small loss, or well-calibrated probabilities, or profitable, or ..)
- Well-calibrated probabilities  
(When the predicted probabilities approximately match the class proportions)
- Confusion matrices  
(We can make a table of the possible outcomes if we have the true labels)
- Quantifying success in classification  
(Specificity, Sensitivity, precision, recall, kappa, profit,...)