Prediction for Machine Learning

Sebastian Funk

Introduction

Logistic Regression

Prediction error

Out of sample prediction

Confusion matrix

Considerations

References

Prediction for Machine Learning

Sebastian Funk

2022-01-24

Motivation

Why do we generate predictions from a model? [McElreath, 2020]

Model design	What are the implications of my model setup?
Model checking	Did the model fit well and how does it behave?
Software validation	If we simulate data from known parameters, can we recover the parameters?
Research design	Applying power analysis to our scientific hypothesis, can we detect what we're looking for?
Forecasting	What do we expect observations beyond the data to look like?

Student activity

In breakout groups, spend five minutes discussing what things you think are important to know about predictions from a model

General case

- Training set of examples (y) and corresponding features (x)
- Train a classifier or regression model, f, such that $y \sim f(x)$
- New features, x^*,
- Predictions are $y^* \sim f(x^*)$
- NB: `predict` (in R) or `.predict()` (in python) not guaranteed to have same output format or arguments

Example - logistic regression

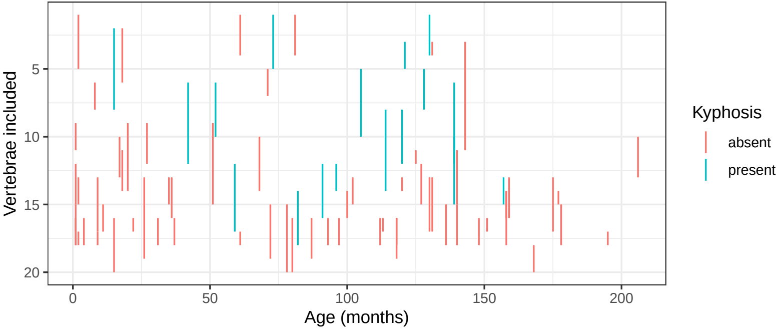
- A simple classifier is logistic regression

$$\mathbb{E}\left[y_i\right] = p_i$$
$$\log\left(\frac{p_i}{1-p_i}\right) = x_i\beta$$
$$\log\left(\frac{p^*_{n+1}}{1-p^*_{n+1}}\right) = x^*_{n+1}\beta$$

- We can either discuss p^*_{n+1} directly, simulate, or allocate to most likely class

Example - logistic regression

- Consider whether a *kyphosis*, a particular spinal curvature, is present after receiving a corrective spinal surgery [Chambers and Hastie, 1992]
- How does $p(\text{kyphosis})$ vary with age and which vertebrae are involved with the surgery (and their first order interactions)?



Example - logistic regression

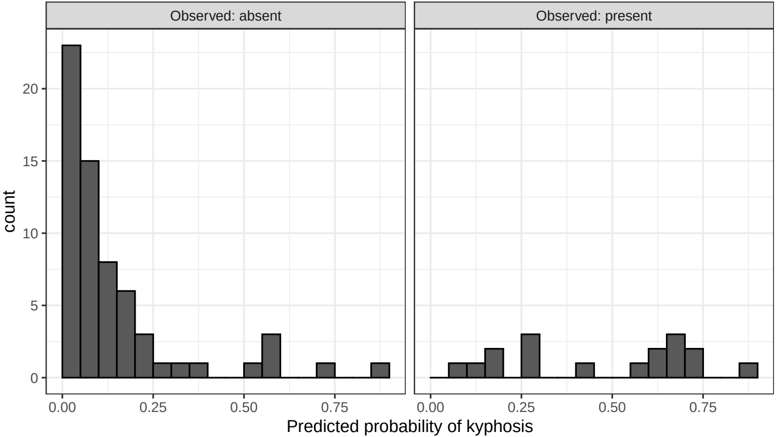
```r
library(tidyverse) # for convenience
library(magrittr)  # for %<>%
data(kyphosis, package = "rpart")
kyphosis %<>% mutate(y = as.numeric(Kyphosis == "present"))

kyph_glm <- glm(data    = kyphosis,
                formula = y ~ Age * Start * Number - Age:Start:Number,
                family  = binomial())
```

And now we predict

```r
kyphosis %<>% mutate(pred = predict(kyph_glm, newdata = ., type = "response"))
```

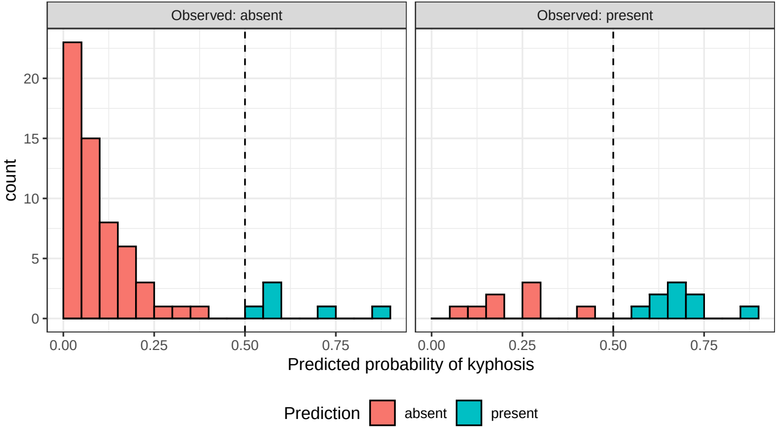
Example - logistic regression

What do our probabilities look like?



Example - logistic regression

Classify with a threshold of p=0.5 for presence/absence.



Example - logistic regression

How often do we predict the right outcome?

Table 1: Confusion matrix: Cross-tabulation of observed (columns) and predicted (rows) kyphosis from GLM

	absent	present
Predicted absent	58	8
Predicted present	6	9

Prediction error

- Recall that the **misclassification error rate** is

$$\frac{1}{n}\sum_{i=1}^{n}\mathcal{I}\left(y_i^* \neq y_i\right).$$

and the **accuracy** is $1 - \text{misclassification error rate}$,

$$\frac{1}{n}\sum_{i=1}^{n}\mathcal{I}\left(y_i^* = y_i\right).$$

- The error rate includes false positives and false negatives
- For our *kyphosis* GLM, we have 6 misclassified absences and 8 misclassified presences, so the error rate is $14/81 = 17\%$.

Student activity

- For our *kyphosis* GLM, we have 6 misclassified absences and 8 misclassified presences, so the error rate is $14/81 = 17\%$.

In breakout groups, spend five minutes discussing if you think the model is good at predicting kyphosis.

Time to try it yourself (in R)

- Omit rows from the NHANES data set that have NA values for diabetes, BMI, age, physical activity, ethnicity, and systolic BP (ensure you only have unique records and no repeat measurements of an individual)
- Draw a subsample of 500 observations from this data set
- Fit either a GLM, Classification tree (including random forest), SVM or KNN with Diabetes as the outcome
- Generate predictions for the data used to train the model
- Convert predictions to class labels and generate a confusion matrix

Out of sample prediction

- We need to assess how well a trained model predicts unseen but known data
- Prediction on training set, x, is **in-sample** prediction
- Prediction on test set, x^*, is **out of sample** prediction
- We hope that performance is similar on these two sets

Motivation

Why do we generate predictions from a model? [McElreath, 2020]

Model design	What are the implications of my model setup?
Model checking	**Did the model fit well and how does it behave?**
Software validation	If we simulate data from known parameters, can we recover the parameters?
Research design	Applying power analysis to our scientific hypothesis, can we detect what we're looking for?
Forecasting	**What do we expect observations beyond the data to look like?**

Out of sample prediction

- Let’s split the *kyphosis* data set in two

```
library(caret)
set.seed(21)
kyph_folds <-
    createFolds(y = kyphosis$Kyphosis,
                k = 2) %>%
    setNames(c('Train', 'Test'))

kyph_train <-
    kyphosis[kyph_folds$Train, ]
kyph_test  <-
    kyphosis[kyph_folds$Test, ]

kyph_glm_train <- glm(
    data    = kyph_train,
    formula = y ~ Age * Start * Number -
        Age:Start:Number,
    family  = binomial())
```

```
## $Train
##          Kyphosis
## pred      absent present
##   absent      30       4
##   present      2       5
##
## $Test
##          Kyphosis
## pred      absent present
##   absent      27       3
##   present      5       5
```

Out of sample prediction

Table 2: Accuracy of GLM fit to training and testing sets of kyphosis data

Set	Accuracy
Train	0.85 (0.71, 0.94)
Test	0.80 (0.64, 0.91)

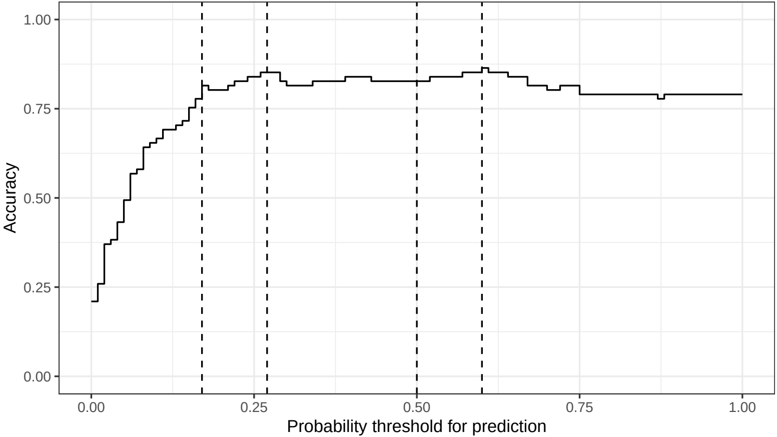
- So we conclude that out of sample prediction and in-sample prediction here are quite similar
- A 50-50 split on 81 observations may not be wise
- Cross-validation will be more useful (tomorrow)

Time to try it yourself (in R)

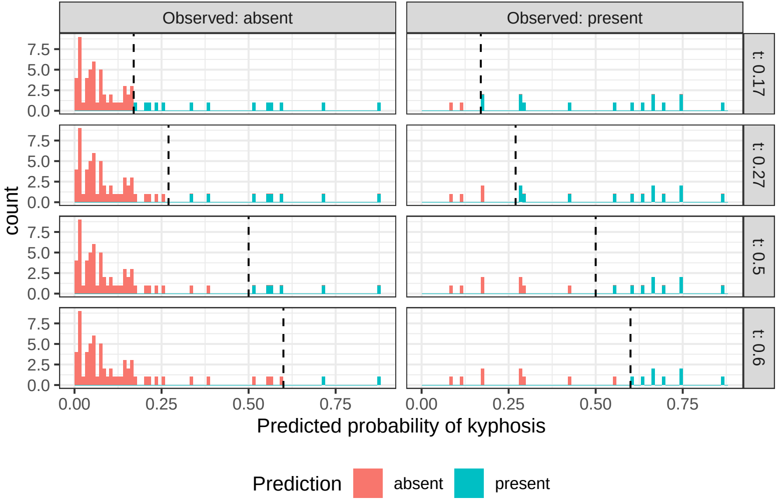
- Draw a subsample of 500 observations from this data set to use as a test set
- Generate predictions for the test set
- Convert predictions to class labels and generate a confusion matrix
- Compare the accuracy of the trained classifier on the test and training sets
    - does the model perform well out of sample when compared to in-sample?
    - does the model perform well out of sample?

Prediction error

- Perhaps a threshold of 0.5 is not the right value
- May wish to find threshold that maximises accuracy, or at least class separation



Prediction error



Confusion matrix

Recall the following tabular representation of true and false positives and negatives for a threshold of 0.5:

	Absent	Present
Predicted absent	TN = 58	FN = 8
Predicted present	FP = 6	TP = 9

For a threshold of 0.17, the confusion matrix is:

	Absent	Present
Predicted absent	TN = 51	FN = 2
Predicted present	FP = 13	TP = 15

TP went from 9 to 15, and FP went from 6 to 13

Confusion matrix

The sensitivity (true positive rate, recall) and specificity (true negative rate) are:

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$
$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

Confusion matrix

Other relevant information we can extract includes:

- Positive predictive value (precision), probability a positive prediction is a true positive
- Negative predictive value, same as for negative predictions

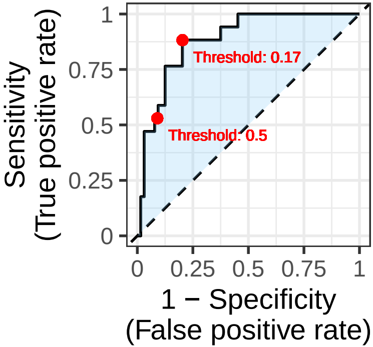
$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

$$\mathrm{NPV} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FN}}$$

Threshold	Sens	Spec	PPV	NPV	Acc
0.5	0.529	0.906	0.600	0.879	0.827
0.17	0.882	0.797	0.536	0.962	0.815

Receiver operating characteristic

- Each point on **ROC** curve corresponds to a threshold between 0 and 1 [Fawcett, 2006]
- Used in biostats to determine where to set threshold for test performance
- Youden's J [Youden, 1950] can be used to find optimum threshold, t, which separates classes

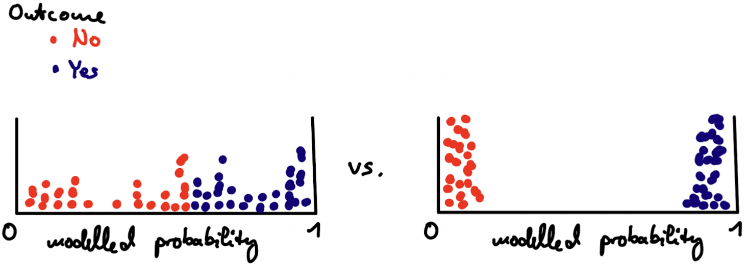


$$\arg\max_t \; (\mathrm{Sens}(t) - (1 - \mathrm{Spec}(t)))$$

Confusion matrix

- Balanced accuracy, average of sensitivity and specificity
  - useful when looking at rare events and classes are imbalanced
- Many of these classifier diagnostics available in `caret::confusionMatrix()` [Kuhn, 2020]
- ROCR implements ROC [Sing et al., 2005] but uses S4 object, so use ggfortify [Tang et al., 2016] to get data frame output
- These techniques hold regardless of what binary classifier has been built

Probabilistic scoring rules



- Even with ROC, still ignore information in the probabilities by using a single threshold value.
- Sometimes I might be interested in how good the predictive *probabilities* are at discrimnating outcomes, rather than assessing the binary classifier
- So-called probabilistic scoring rules take into account the whole range of predictions generated and compare them to the data [Gneiting and Raftery, 2007]

Examples of scoring rules

- *Log score ("Log loss")*

$$L = \sum_{i=1}^{N} \log x_i$$

where x_i is the predicted probability p_i of the observed outcome o_i

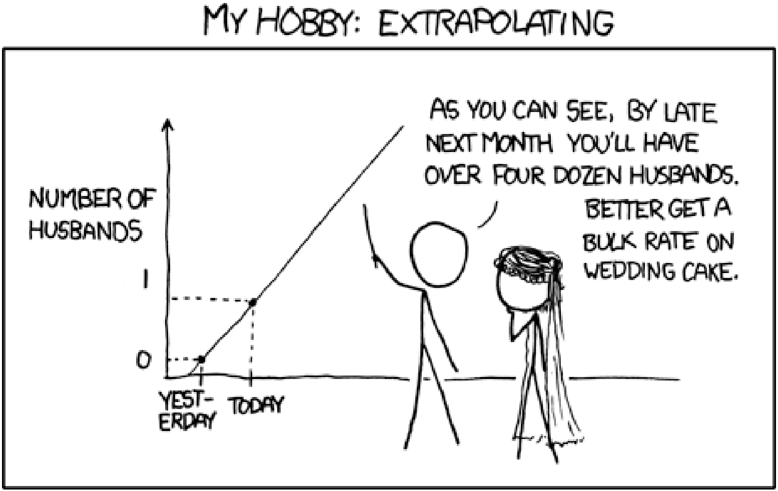
$$x_i = \begin{cases} p_i, & \text{if } o_i = 1 \\ (1 - p_i), & \text{if } o_i = 0 \end{cases}$$

- *Brier score*

$$B = \frac{1}{N} \sum_{i=1}^{N} (p_i - o_i)^2$$

Considerations for predicting

- Is the prediction out of sample?
  - how similar is x^* to all x?
- How independent are my data?
  - is there any overlap in the testing and training sets?
- Has my model been overfit?
- Do my predictions involve trends in the x that will continue to occur?
- Do my predictions give guidance one way or another as to a recommended course of action?
- How confident can we be in the predictions?


MY HOBBY: EXTRAPOLATING
NUMBER OF HUSBANDS
1
0
YEST-ERDAY
TODAY
AS YOU CAN SEE, BY LATE NEXT MONTH YOU'LL HAVE OVER FOUR DOZEN HUSBANDS. BETTER GET A BULK RATE ON WEDDING CAKE.

Further reading

- For a good discussion on the distinction between prediction, estimation and attribution in statistical and machine learning, see Efron [2020] and Shmueli [2010]
- Provost [2000] discusses issues in applying ML techniques to data sets with extreme imbalances in class membership
- Lum and Isaac [2016] considers the evidence, and the social consequences, of the use of biased data in training models for crime prediction
- How good are your beliefs? Part 1: Scoring Rules is a good nontechnical introduction to probabilistic scoring rules.

John M Chambers and Trevor J Hastie. *Statistical models in S*. Wadsworth & Brooks/Cole New York, Ch, 1992.

Bradley Efron. Prediction, estimation, and attribution. *Journal of the American Statistical Association*, 115(530):636–655, April 2020. doi: 10.1080/01621459.2020.1762613. URL https://doi.org/10.1080/01621459.2020.1762613.

Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, June 2006. URL https://doi.org/10.1016/j.patrec.2005.10.010.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

Max Kuhn. *caret: Classification and Regression Training*, 2020. URL https://CRAN.R-project.org/package=caret. R package version 6.0-86.

Kristian Lum and William Isaac. To predict and serve? *Significance*, 13(5):14–19, October 2016. URL https://doi.org/10.1111/j.1740-9713.2016.00960.x.

Richard McElreath. *Sampling to simulate prediction. From: Statistical rethinking: A Bayesian course with examples in R and Stan*, chapter 3.3. CRC press, 2020.

Foster Provost. Machine learning from imbalanced data sets 101. In *Proceedings of the AAAI '2000 workshop on imbalanced data sets*, volume 68, pages 1–3. AAAI Press, 2000.

Galit Shmueli. To explain or to predict? *Statistical Science*, 25(3): 289–310, August 2010. doi: 10.1214/10-sts330. URL https://doi.org/10.1214/10-sts330.

T. Sing, O. Sander, N. Beerenwinkel, and T. Lengauer. ROCR: visualizing classifier performance in r. *Bioinformatics*, 21(20): 7881, 2005. URL http://rocr.bioinf.mpi-sb.mpg.de.

Yuan Tang, Masaaki Horikoshi, and Wenxuan Li. ggfortify: Unified interface to visualize statistical result of popular r packages. *The R Journal*, 8, 2016. URL https://journal.r-project.org/.

William J Youden. Index for rating diagnostic tests. *Cancer*, 3(1): 32–35, 1950.