Prediction

Dan A. Greenberg

R Users/SIO/March 11 2021

X predicts Y...

Claims of prediction are widespread in ecology & evolution

Rarely, however, are predictive tests
actually formally conducted

A seemingly good model (fits and explains variance well) may still fail to actually make accurate predictions(!)

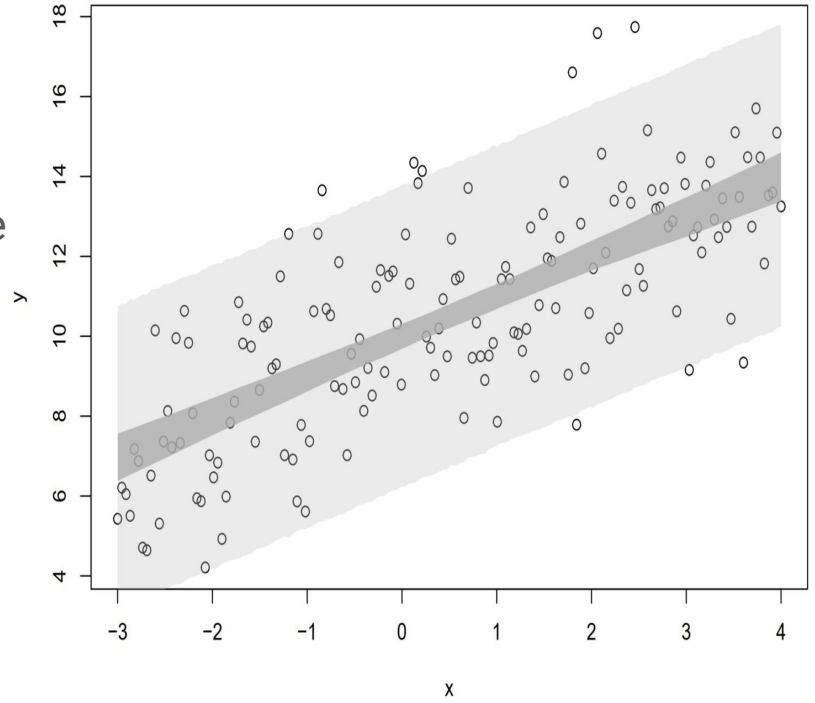
Can test this for *any* model and data

Nostradamus

A quick aside: prediction intervals

Confidence (credible) interval - A range of values for an unknown parameter (eg. mean effect) representing the long-run frequency, eg. 90% of the time the sampled value will fall within this band

Prediction interval - A range of values that will contain a future sample with X% probability, eg. 90% of the time a new sample will fall within this band.



Estimating prediction intervals

One can probabilistically estimate prediction intervals for your model - including a number of different uncertainties in your parameter estimates

For a simple linear model...

$Y = B0 + B1*x1 + \varepsilon$, where $\varepsilon \sim N(0, \sigma)$

Uncertainty arises in estimates of the parameters: B0, B1, and ε (residual variation). This uncertainty is also not necessarily *independent*.

Variance & Covariance

Must account for correlations in the parameter estimates themselves - can do this by extracting the variance-covariance matrix:

Eg.

$\sigma 1$	$\rho\sigma 1\sigma 2$
$\rho\sigma 1\sigma 2$	$\sigma 2$

Now can simulate draws of each parameter (eg. B0) based on its independent variance (eg, $\sigma 1$) and the covariance with other parameters (eg. $\rho\sigma 1\sigma 2$)

In R...

```r
b0<- 10
b1<- 1
x<- seq(-3,4,length.out=160)
y<- b0+b1*x+rnorm(160,0,2)
var(y)
plot(y~x)
cor(x,y)
data<- data.frame(x=x,y=y)
l0<- lm(y ~ x, data=data)
betas<- coef(l0)
vcv<- vcov(l0)
pars.resamp<- MASS::mvrnorm(500, mu = betas, Sigma = vcv)
```

Generating predictions

To predict, one needs options from out-of-sample (ie. data not used to parameterize your model - this is also termed a testing dataset)

Rarely do we have truly novel data to test upon (this would require a completely different dataset for a true naive test)

But, we can approximate some facsimile of novelty - through cross-validation

Cross-validation

A simple exercises that partitions your data into two sets - a training dataset and a testing (or out-of-sample) dataset

Different kinds of cross-validation:

1. **Leave-one-out**: Remove a single data-point and attempt to predict the response with your model built on all the remaining data
2. ***K*-fold**: Randomly remove *K*% (eg. 10-fold = 10%) of your dataset to retain as the out-of-sample prediction
3. Other forms -- eg. removing specific ‘blocks’, more on this later

Easy to do in R using the *sample* function

In R...

```r
p2<- NA
for(i in 1:500){
    row_sample<- sample(nrow(data),round(nrow(data)*0.1)) #randomly sample 10% of dataset
    test<- data[row_sample,] #Keep this subset for testing
    train<- test[-row_sample,] #Drop this subset for model training

    lt<- lm(y ~ x+grp, data=train) #fit model to training set
    preds<- predict(lt,newdata=test) #make predictions from that model to the testing set
    p2[i]<- cor(test$y,preds)^2 #Estimate prediction accuracy (obs. vs. predicted)

}
hist(p2)
abline(v=median(p2),lty=5)
```

Measuring predictive accuracy

To claim that X predicts Y - we must measure some aspect of predictive *accuracy*

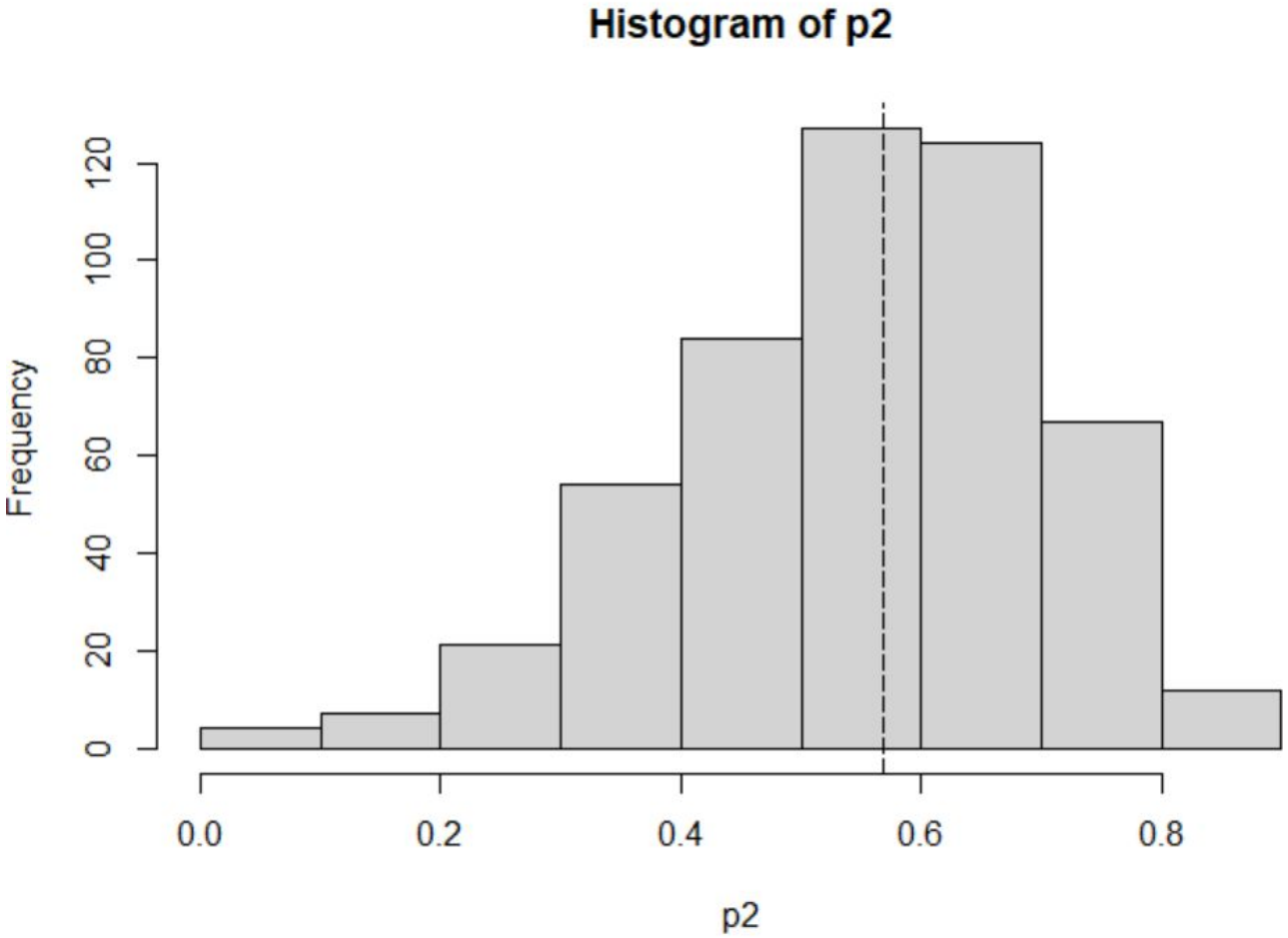
What is this will depend on your data:

Continuous data (eg. -inf, +inf; 0, inf; etc.)

- -Mean Squared Error (MSE), RMSE, R^2 from observed vs. predicted

Categorical data (eg. 0,1; 0,..,N; etc.)

- -Accuracy (TP + TN/P + N)
- -AUC from Receiver Operating Characteristic curves


Histogram of p2
Frequency
0
20
40
60
80
100
120
0.0
0.2
0.4
0.6
0.8
p2

In R...

```r
p2<- NA
for(i in 1:500){
    row_sample<- sample(nrow(data),round(nrow(data)*0.1)) #randomly sample 10% of dataset
    test<- data[row_sample,] #Keep this subset for testing
    train<- test[-row_sample,] #Drop this subset for model training

    lt<- lm(y ~ x+grp, data=train) #fit model to training set
    preds<- predict(lt,newdata=test) #make predictions from that model to the testing set
    p2[i]<- cor(test$y,preds)^2 #Estimate prediction accuracy (obs. vs. predicted)

}
hist(p2)
abline(v=median(p2),lty=5)
```

Thinking carefully about your predictions...

Many of us work with data with various intraclass correlation structures

Eg. If your data is across different species, sampling sites, seasons, etc.

Perform cross-validation on entire groups if you want to predict for sampling a new species, site, etc.

```r
tau_a<- rep(rnorm(20,0,1.5),8) #group variation in intercept
tau_b<- rep(rnorm(20,0,0.05),8) #group variation in slope
grp<- rep(seq(1,20),8) #group id

y<- b0+tau_a+(b1+tau_b)*x+rnorm(160,0,1.5) #simulate with group differences
var(y)
plot(y~x,col=as.factor(grp))
cor(x,y)
l0<- lm(y ~ x, data=data) #fit full model
summary(l0)

p2<- NA
for(i in 1:500){
  grp_sample<- sample(20,1) #randomly sample 1 of the 20 groups to drop
  test<- data[grp %in% grp_sample,] #Keep this subset for testing
  train<- data[grp %notin% grp_sample,] #new training dataset
  row_sample<- sample(nrow(train),8) #also randomly drop 8 rows from remaining groups
  test<- rbind(test,train[row_sample,])
  train<- train[-row_sample,]

  lt<- lm(y ~ x+grp, data=train) #fit model to training set
  preds<- predict(lt,newdata=test) #make predictions from that model to the testing set
  p2[i]<- cor(test$y,preds)^2 #predictive accuracy

}
hist(p2)
abline(v=median(p2),lty=5)
```

Final thoughts on applications

If your aim is predicting some outcome - then it’s key to at least do a cross-validation (in my opinion)

Helps you quantify the uncertainty in forecasts and verifies where your models fit well or fall apart

Same principles can apply to other applications - eg. sensitivity analyses, boot-strapped parameters, etc.