Model accuracy

Katie Schuler

2023-10-29

1 Model accuracy basics

We've selected a model (model selection) and fit a model to a set of data (model fitting). One question we might want to ask next is how well does this model describe the data (**model accuracy**)?

- We can visualize our data and the model fit to get a sense of how accurate the model is. But we also want a way to quantify model accuracy – some metric by which to determine whether a model is useful, or how it compares to other models.
- Last week we learned about one metric of model "goodness", **Sum of Squared Error (SSE)**. We could certainly quantify our model accuracy with SSE, but it would be difficult to interpret since it depends on the units of the data.
- Today we'll learn about another metric, R^2 which is easier to interpret and independent of units. R^2 quantifies the percentage of variance in our response variable that is explained by our model.

2 Coefficient of determination, R^2

The coefficient of determination, R^2, quantifies the proportion of **variance** in the response variable, y, that is explained by the model. Since you've already learned about **Sum of Squared Error (SSE)** as a way to quantify how well a model fits the data, you already have the tools to understand R^2.

To obain R^2 for a model, we compare the SSE of our model with the SEE of the simplest possible model: `y ~ 1` (the mean of the observed y values). We call this simple model the **reference model** in the R^2 equation.

The equation for R^2 is:

- $R^2 = 100 \times (1 - \frac{\sum_{i=1}^{n}(y_i - m_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2})$

Which is the same as saying:

- $R^2 = 100 \times (1 - \frac{SSE_{model}}{SSE_{reference}})$

Sometimes you will also see it expressed like this:

- $R^2 = 100 \times (1 - \frac{unexplained\ variance}{total\ variance})$

Which helps us appreciate what the equation is doing in terms of SSE:

- The *total variance* (denominator) is quantified as the sum of squared error in which we substract the mean from the data (the simplest model's prediction): $\sum_{i=1}^{n}(y_i - \overline{y})^2$. In words, take each y value and subtract it from the mean y value, square it, then add them all up.
- The *unexplained variance* (numerator) is quantified as the sum of squared error in which we subtract the model value from the data (residuals): $\sum_{i=1}^{n}(y_i - m_i)^2$. In words, take each y value and subtract it from the model value (the model's prediction) for that data point, square it, then add them all up.

We then subtract the proportion $\frac{unexplained\ variance}{total\ variance}$ from 1 to get the proportion of variance that *is* explained, and then we multiply by 100 to turn it into the percent of variance explained.

- **There is an upper bound of 100%**: the situation where the model explains all the variance (it matches the data exactly)
- **There is technically no lower bound**, since models can be arbitrarily bad. **0%** indicates the model explains none of the variance (it predicts the mean of the data but nothing else)

3 R^2 overestimates model accuracy

One thing we can ask is how well the model describes our specific sample of data. But the question we actually want to answer is *how well does the model we fit describe the population we are interested in.*

- The problem is that we usually only have access to the sample we've collected and R^2 tends to **overestimate** the accuracy of the model on the population. In other words, the R^2 of the model we fit on our sample will be larger than the R^2 of the model fit to the population.
- Further, the population is (usually) unknown to us. To quantify the **true accuracy** of a fitted model – that is, how well the model describes the population, not the sample we collected – we can use a technique called **cross-validation**.

Before we learn about cross-validation, let's first try to gain further conceptual understanding of *why* R^2 tends to overestimate model accuracy.

4 Overfitting

When you fit a model to some sample of data, there is always a risk of **overfitting**. As the modeler, you have the freedom to fit your sample data better and better (you can add more and more terms, increasing the R^2 value). But you need to be careful not to fit the sample data *too* well.

- This is because any given set of data contains not only the true, underlying patterns we are interested in (the **true model** or **signal**), but also random variation (**noise**). Fitting the sample data too well means we fit not only the signal but also the noise in the data.
- An overfit model will perform really well on the data it has been trained on (the sample) — we can even fit the sample perfectly if we add enough terms! - but an overfit model will be bad at predicting new, unseen values. Image we collect an additional data point drawn from the population. An overfit model would predict this point poorly!
- Our goal is to find the **optimal** fitted model – the one that gets as close to the true model as possible without overfitting. But we have no way of knowing which part of the data we sampled is signal and which part is noise. So, we use cross-validation to help identify overfitting.

5 Model complexity

In the lecture on model specification, we briefly mentioned that we would also want to take into consideration the **complexity** of the model. Simple models are easier to interpret but may not capture all complexities in the data, while complex models can suffer from overfitting the data or be difficult to interpret. Let's expand on this in the context of model accuracy.

- **Complex models** have the potential to describe many kinds of functions, and the *true model* — the model that most accurately describes the population we sampled our data from — could be among them. However, complex models have a lot free parameters to estimate (by definition, that's what makes them complex!), which makes it more difficult to obtain stable parameter estimates with small samples sizes or noisy data.
- **Simple models** are limited in the types of functions they can describe, so they may not approximate the true model very accurately. However, they have fewer free parameters, which makes it easier to obtain stable parameter estimates with small sample sizes or noisy data.

- We have no way of knowing *a priori* whether a simple or complex model will be more accurate for a given dataset. It depends on many things, including the data we have, the underlying relationships, and our research questions. Luckily, we can use cross-validation to find out, trying different models and quantify each model's accuracy.

6 Cross-validation

Remember from above, the question we actually want to answer with R^2 is not how well does the model we fit describe the sample we collected, but *how well does the model we fit describe the population we are interested in*. But R^2 on the sample will tend to overestimate the model's accuracy on the population. To estimate the accuracy of the model on the population, we need to use a simple but powerful technique called **cross-validation**. Given a sample of data, there are 3 simple steps to any cross-validation technique:

1. Leave some data out
2. Fit a model (to the data kept in)
3. Evaluate the model on the left out data (e.g. R^2)

There are many ways to do cross-validation — reflecting that there are many ways we can leave some data out — but they all follow this general 3-step process. We'll focus on two common approaches in this class:

- In **leave-one-out** cross-validation, we leave out a single data point and use the fitted model to predict that single point. We repeat this process for every data point, then evaluate each model's prediction on the left out points (we can use R^2!).
- In k**-fold** cross-validation, instead of leaving out a single data point, we randomly divide the dataset into k parts and use the fitted model to predict that *part*. We repeat this process for every part, then evaluate each model's prediction on the left out parts (again, we can use R^2!).

How do we decide which cross-validation approach to use? There are two trade-offs to consider:

(1) **How many iterations** do we want to do? The more iterations, the more reliable our accuracy estimate will be. But the more iterations, the more computational resources are required.
(2) **How much data** do we want to use for each part? The more data we use to fit the model, the more accurate the model will be and the more stable the parameter estimates will be. But the more data we use in to estimate reliability, the more reliable our accuracy estimate will be.

- For example, in leave-one-out cross-validation we use a lot of iterations (one for each data point), so we need a lot of computational resources, but we get to use almost all the data to fit our model (all but one point!) and all the data to calculate R^2.
- Keep in mind that the parameter estimates we obtain on each iteration will be different, because they depend on both the model selected (stays the same each iteration) and the data we fit with (changes each iteration). So the R^2 we compute via cross-validation really reflects an estimate of our model's accuracy when fitted to a particular amount of data.

7 Other methods

There are other ways to evaluate models beyond cross-validaiton.

One common way is using an **F-test** to determine whether a more complex model produces a significantly better fit than a simpler one. This approach only applies for *nested models*, which just means that one model is a simpler version of another more complex one.

You may also encounter **AIC (Akaike Information Criterion)** and **BIC (Bayesian Information Criterion)**, for example, which are parametric approaches that attempt to compare different models and find the optimal fit (helping you avoid overfitting and excessively complex models).

- In general AIC considers how well the model fits the data, the number of parameters, and the sample size (there is a penalty for more complex models); BIC is similar but has a stronger penalty for complex models (so will inherently favor simpler models).
- **We'll focus on cross-validation** in this class, because it makes fewer assumptions than metrics like AIC/BIC and is simpler to understand conceptually. But we'll also show you the **F-test** approach, since it's widely used in the sciences.

8 F-test (via `anova()`)

The F-test is closely related to R^2. When comparing a simpler model to a more complex one, the **change in** R^2 can be evalutated using an F-test to see if adding predictors significantly improves model fit. (often expresse as ΔR^2) Recall that, for R^2, when we compared SSE_{model} (the sum of squared error of our model) to $SSE_{reference}$ (the sum of squared error of the intercept-only model), we noted that $SSE_{reference}$ is always going to be greater than SSE_{model}. But what we actually want to know is whether it is *significantly* greater. Said another way, we want to know whether adding terms to the model significantly improve the model's ability to explain the response variable.

Let R^2_{simple} be the R^2 of the simpler model and $R^2_{complex}$ be the R^2 of the more complex model. The change in R^2 (also called ΔR^2) is:

- $\Delta R^2 = R^2_{complex} - R^2_{simple}$

We can then compute the F-statistic to determine if ΔR^2 is significant.

- $F = \frac{\Delta R^2 / p}{(1 - R^2_{complex}/(n-k-1))}$

Where:

- p is the number of additional predictors in the complex model
- n is the total sample size
- k is the number of predictors in the complex model

We can understand the numerator and denominator of this equation in the following way:

- The numerator represents the increase in *explained variance* per additional predictor.
- The denominator represents the remaining *unexplained variance*, adjusted for sample size and the complexity of the model.

In R, we can perform this model comparison via and F-test via a call to `anova()`:

```
model_int <- lm(RTlexdec ~ 1, english)
model_freq <- lm(RTlexdec ~ WrittenFrequency, english)
model_freqage <- lm(RTlexdec ~ WrittenFrequency + AgeSubject, english)
model_freqagelength <- lm(RTlexdec ~ WrittenFrequency + AgeSubject + LengthInLetters, englis

anova(model_int, model_freq, model_freqage, model_freqagelength)
```

```
Analysis of Variance Table

Model 1: RTlexdec ~ 1
Model 2: RTlexdec ~ WrittenFrequency
Model 3: RTlexdec ~ WrittenFrequency + AgeSubject
Model 4: RTlexdec ~ WrittenFrequency + AgeSubject + LengthInLetters
  Res.Df     RSS Df Sum of Sq         F  Pr(>F)
1   4567 112.456
2   4566  91.194  1    21.261 2772.1326 < 2e-16 ***
3   4565  35.053  1    56.141 7319.9087 < 2e-16 ***
4   4564  35.004  1     0.049    6.3563 0.01173 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

If the F-statistic is large, it suggests that the additional predictors in the complex model significantly improve model fit. To help you decide, `anova()` returns a p-value. You can understand this p-value as asking: how likely it is to observe this value of F if we randomly added this many predictors to our model?

9 Back to model selection

Building models is itself an iterative process: we can use model accuracy obtained via cross-validation to determine which model to select (as a way to find the elusive optimal model fit).

Cross-validation seems preferable, as it makes fewer assumptions than these approaches and is conceptually simpler. However, a drawback of cross-validation is that it is computationally intensive.

Beyond model accuracy, there are other practical things one might want to consider when selecting a model, such as ease of interpretation and availability of resources (the data you can collect, the computing power you have, etc.)

10 Further reading