Continuous soil attribute modeling and mapping: Goodness of fit and model validation.

Soil Security Laboratory

2017

The implementation of some of the most commonly used model functions used for digital soil mapping will be covered in this chapter. Before this is done however, some general concepts of model validation are covered.

1 Model validation

Essentially, whenever we train or calibrate a model, we can then generate some predictions. The question one needs to ask is how good are those predictions? Generally, we confront this question by comparing observed values with their corresponding predictions. Some of the more common "quality" measures are the root mean square error (RMSE), bias, coefficient of determination or commonly the R^2 value, and concordance. You will also find in the digital soil mapping and general statisitcal literature various other model assessment statisitcs. The RMSE is defined as:

$$RMSE = \sqrt[2]{(\frac{\sum_{i=1}^{n}(obs_i - pred_i)^2}{n})} \tag{1}$$

where *obs* is the observed soil property, *pred* is the predicted soil property from a given model, and n is the number of observations i. Bias, also called the mean error of prediction and is defined as:

$$bias = \frac{\sum_{i=1}^{n} pred_i - obs_i}{n} \tag{2}$$

The R^2 is evaluated as the square of the sample correlation coefficient (Pearson's) between the observations and their corresponding predictions. Pearson's correlation coefficient r when applied to observed and predicted values is defined as:

$$r = \frac{\sum_{i=1}^{n}(obs_i - \overline{obs})(pred_i - \overline{pred})}{\sqrt[2]{\sum_{i=1}^{n}(obs_i - \overline{obs})^2}\sqrt[2]{\sum_{i=1}^{n}(pred_i - \overline{pred})^2}} \tag{3}$$

The R^2 measures the precision of the relationship (between observed and

predicted). Concordance, or more formally — Lin's concordance correlation coefficient (Lin, 1989), on the other hand is a single statistic that both evaluates the accuracy and precision of the relationship. It is often referred to as the goodness of fit along a 45 degreee line. Thus it is probably a more useful statistic than the R^2 alone. Concordance ρ_c is defined as:

$$\rho_c = \frac{2\rho\sigma_{pred}\sigma_{obs}}{\sigma^2_{pred} + \sigma^2_{obs} + (\mu_{pred} - \mu_{obs})^2} \tag{4}$$

where μ_{pred} and μ_{obs} are the means of the predicted and observed values respectively. σ^2_{pred} and σ^2_{obs} are the corresponding variances. ρ is the correlation coefficient between the predictions and observations.

1.1 Model goodness of fit

So lets fit a simple linear model. We will use the `soil.data` set used before in the introductory to `R` chapter. First load the data in. We then want to regress CEC content on clay (also be sure to remove as NAs).

```
library(ithir)
library(MASS)
```

```
## Warning:  package 'MASS' was built under R version 3.2.5
```

```
data(USYD_soil1)
soil.data <- USYD_soil1
mod.data <- na.omit(soil.data[, c("clay", "CEC")])
mod.1 <- lm(CEC ~ clay, data = mod.data, y = TRUE, x = TRUE)
mod.1
```

```
##
## Call:
## lm(formula = CEC ~ clay, data = mod.data, x = TRUE, y = TRUE)
##
## Coefficients:
## (Intercept)         clay
##      3.7791       0.2053
```

You will recall that this is the same model that we fitted during the introduction to `R` chapter. What we now want to do is evaluate some of the model quality statistics that were just described. Conveniently, these are available in the `goof` function in the `ithir` package. We will use this function a lot during this chapter, so it might be useful to describe it. `goof` takes four inputs. A vector of `observed` values, a vector of `predicted` values, a logical choice of whether an output plot is required, and a character input of what type of output is required. There are number of possible goodness of fit statistics that can be requested, with only some being used frequently in digital soil mapping projects. Therefore setting the `type` parameter to `''DSM''` will output only the R^2, RMSE, MSE, bias and concordance statistics as these are most most relevant to DSM. Additional statitistics can be returned if `''spec''` is specified for the `type` parameter

```
goof(observed = mod.data$CEC, predicted = mod.1$fitted.values, type = "DSM")
```

```
##          R2 concordance      MSE     RMSE bias
## 1 0.4173582   0.5888521 14.11304 3.756733    0
```

You may wish to generate a plot in which case you would set the `plot.it` logical to `TRUE`.

This model `mod.1` does not seem to be too bad. On average the predictions are 3.75 cmol $(+)/kg$ off the true value. The model on average is neither over- or under-predictive, but we can see that a few high CEC values are influencing the concordance and R^2. This outcome may mean that there are other factors that influence the CEC, such as mineralogy type.

1.2 Model validation

Above we performed goodness of fit assessment of the `mod.1` model. Usually it is more appropriate however to validate a model using data that was not included for model fitting. Model validation has a few different forms. For completley unbiased assessments of model quality it is ideal to have an additional data set that is completely independent of the model data. It is recommended that a desgin based random sampling from the target area be conducted, to which there are a few types such as simple random sampling and stratified simple random sampling. Further information regarding sampling, sampling desgings, their formulation and the relative advantages and constraints of each are described in de Gruijter et al. (2006). Usually from an operational perspective it is difficult to arrange the additional costs of organising and implementing some sort of probability sampling for determinig unbiased model quality assessment. The alternative is to perform some sort of data sub-setting, such that with a data set we split it into a set for model calibration and another set for validation. This type of procedure can take different forms: the two main ones being random-hold back and leave-one-out-cross-validation (LOCV). Random-hold back (or sometimes k-fold validation) is where we may sample a data set of some pre-determined proportion (say 70%) for which is used for model calibration. We then validate the model using the other 30% of the data. For k-fold validation we divide the data set into equal sized partitions or folds, with all but one of the folds being used for the model calibration, the remaining fold is used for validation. We could repeat this k-fold process a number of times, each time using a different random sample from the data set for model calibration and validation. This allows one to efficently derive distributions of the validation statisics as a means of assessing the stability and sensitivity of the models and parameters.

LOCV involves a little more computation such that if we had n number of data, we would subset *n-1* of these data, and fit a model. Using this model we would make a prediction for the single data that was left out of the model (and save the residual). This is repeated for all n. LOCV would be undertaken when there are very few data to work with. When we can sacrifice a few data points, the random-hold back or k-fold cross-validation procedure would be

acceptable.

When we are validating trained models with some sort of data sub-setting mechanism, always keep in mind that the validation statitsics will be biased. As Brus et al. (2011) explains, the sampling from the target mapping area to be used for DSM is more often than not from legacy soil survey, to which would not have been based on a probability sampling design. Therefore, that sample will be biased i.e not a true representation of the total population. Even though we may randomly select observations from the legacy soil survey sites, those vlidation points do not become a probaility sample of the target area, and consequently will only provide biased estimates of model quality. Thus an independent probaility sample is required. Further ideas on the statisical valdation of models can be found in Hastie et al. (2001).

So lets implement some of the validation techniques in `R`. We will use the same data as before i.e regressing CEC with clay content. First we will do the random-back validation using 70% of the data for calibration. A random sample of the data will be performed using the `sample` function.

```
set.seed(123)
training <- sample(nrow(mod.data), 0.7 * nrow(mod.data))
training
```

```
##   [1]  42 115  59 127 134   7  74 125  77  63 131  62  91 138  14 118  32
##  [18]   6 146 122 113  87  80 123 124  86  66  71  35  18 112 104  79  90
##  [35]   3  54  84  24 136  25  16  44 105  38 106  15 109  47  27 110   5
##  [52]  43  76  12  52  19  93  68 114  33  58   9 139  23  67  37  65 143
##  [69] 135  34 121  48  53   1 108 102  98  95 100   8  17 145  70  50 141
##  [86]  64  60 140  92  10  82  36 142  72 120  57  40  96  83 107  28 101
```

These values correspond to row numbers which will correspond to the row which we will use for the calibration data. We subset these rows out of `mod.data` and fit a new linear model.

```
mod.rh <- lm(CEC ~ clay, data = mod.data[training, ], y = TRUE, x = TRUE)
```

So lets evaluate the calibration model with `goof`:

```
goof(predicted = mod.rh$fitted.values, observed = mod.data$CEC[training])
```

```
##          R2 concordance      MSE     RMSE bias
## 1 0.4457907   0.6158071 12.31952 3.509917    0
```

But we are more interested in how this model performs when we use the validation data. Here we use the `predict` function to predict upon this data.

```
mod.rh.V <- predict(mod.rh, mod.data[-training, ])
goof(predicted = mod.rh.V, observed = mod.data$CEC[-training])
```

```
##          R2 concordance      MSE     RMSE       bias
## 1 0.3591283   0.5208349 18.35828 4.284656 -0.5355242
```

So the model is not as good as we first imagined.When we validate a model with an external data set, it is quite normal that the model will not perform

nearly as well as when using calibration data. Set the `plot.it` parameter to `TRUE` and re-run the script above and you will see a plot like Figure 1.

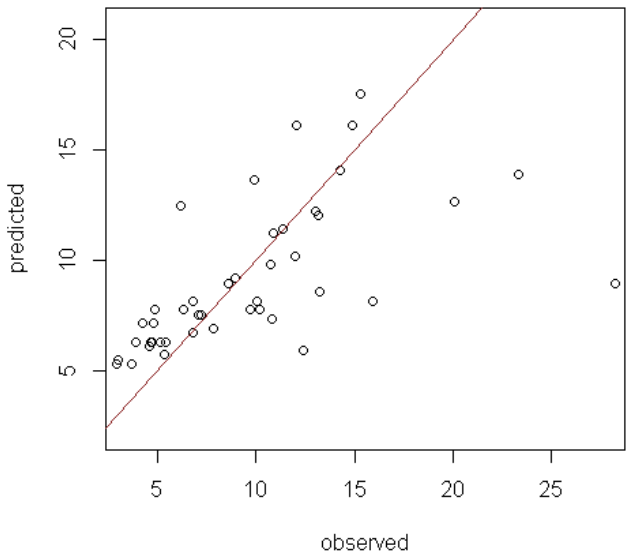

Figure 1: Observed vs. predicted plot of CEC model (validation data set) with line of concordance (red line).

In fact the `mod.rh` model does not appear to perform too bad after all. A few of the high observed values contribute greatly to the validation diagnostics. A couple of methods are available to assess the sensitivity of these results. The first is to remove what could potentially be outliers from the data. The second is to perform a sensitivity analysis such as bootstrapping where we iterate the data sub-setting procedure and evaluate the validation statistics each time to get a sense how much they vary.

At the most basic level, LOCV involves the use of a looping function or `for` loop. We have not really covered `for` loops yet, but essentially they can be used to great effect when we want to perform a particular analysis over-and-over. For example with LOCV, for each iteration or loop we take a subset of *n-1* rows and fit a model to them, then use that model to predict for the point left out of the calibration. Computationally it will look something like this:

```
looPred <- numeric(nrow(mod.data))
for (i in 1:nrow(mod.data)) {
    looModel <- lm(CEC ~ clay, data = mod.data[-i, ], y = TRUE, x = TRUE)
    looPred[i] <- predict(looModel, newdata = mod.data[i, ])
}
```

The `i` here is the counter, so for each loop it increases by 1 until we get to

the end of the data set. As you can see, we can index the `mod.data` using the `i`, meaning that for each loop we will have selected a different calibration set. On each loop, the prediction on the point left out of the calibration is made onto the corresponding row position of the `looPred` object. Again we can assess the performance of the LOCV using the `goof` function.

```
goof(predicted = looPred, observed = mod.data$CEC)
```

```
##          R2 concordance      MSE     RMSE        bias
## 1 0.4025255   0.5790589 14.47653 3.804804 0.005758669
```

LOCV will generally be less sensitive to outliers, so overall these external validation results are not too different to those when we performed the internal validation. Make a plot of the LOCV results to visually compare against the internal validation.

References

Brus, D., B. Kempen, and G. Heuvelink
2011. Sampling for validation of digital soil maps. *European Journal of Soil Science*, 62(3):394–407.

de Gruijter, J., D. Brus, M. Bierkens, and M. Knotters
2006. *Sampling for Natural Resource Monitoring*. Berlin Heidelberg: Springer-Verlag.

Hastie, T., R. Tibshirani, and J. Friedman
2001. *The Elements of Statistical Learning*. New York, NY: Springer.

Lin, L. I.
1989. A concordance correlation coeficient to evaluate reproducibility. *Biometrics*, 45:255–268.