Validation and accuracy measurements in SITS

Rolf Simoes *National Institute for Space Research (INPE), Brazil*
Gilberto Camara *National Institute for Space Research (INPE), Brazil*
Pedro R. Andrade *National Institute for Space Research (INPE), Brazil*
Victor Maus *University of Vienna*

This vignette presents the validation and accuracy measures available in the SITS package.

Validation techniques

Validation is a process undertaken on models to estimate some error associated with them, and hence has been used widely in different scientific disciplines. Here, we are interested in estimating the prediction error associated to some model. For this purpose, we concentrate on the *cross-validation* approach, probably the most used validation technique [Hastie et al., 2009].

To be sure, cross-validation estimates the expected prediction error. It uses part of the available samples to fit the classification model, and a different part to test it. The so-called *k-fold* validation, we split the data into k partitions with approximately the same size and proceed by fitting the model and testing it k times. At each step, we take one distinct partition for test and the remaining $k-1$ for training the model, and calculate its prediction error for classifying the test partition. A simple average gives us an estimation of the expected prediction error.

A natural question that arises is: *how good is this estimation?* According to Hastie et al. [2009], there is a bias-variance trade-off in choice of k. If k is set to the number of samples, we obtain the so-called *leave-one-out* validation, the estimator gives a low bias for the true expected error, but produces a high variance expectation. This can be computational expensive as it requires the same number of fitting process as the number of samples. On the other hand, if we choose $k = 2$, we get a high biased expected prediction error estimation that overestimates the true prediction error, but has a low variance. The recommended choices of k are 5 or 10 [Hastie et al., 2009], which somewhat overestimates the true prediction error.

`sits_kfold_validate()` gives support the k-fold validation in `sits`. The following code gives an example on how to proceed a k-fold cross-validation in the package. It perform a five-fold validation using SVM classification model as a default classifier. We can see in the output text the corresponding confusion matrix and the accuracy statistics (overall and by class).

```
# perform a five fold validation for the "cerrado_2classes" data set
# Random Forest machine learning method using default parameters
```

```
prediction.mx <- sits_kfold_validate(cerrado_2classes,
                                     folds = 5,
                                     ml_method = sits_rfor())
# prints the output confusion matrix and statistics
sits_conf_matrix(prediction.mx)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Cerrado Pasture
##    Cerrado     394      14
##    Pasture       6     332
##
##           Accuracy : 0.9732
##             95% CI : (0.9589, 0.9835)
##
##              Kappa : 0.946
##
##  Prod Acc  Cerrado : 0.9850
##  Prod Acc  Pasture : 0.9595
##  User Acc  Cerrado : 0.9657
##  User Acc  Pasture : 0.9822
##
```

Comparing different validation methods

One useful function in SITS is the capacity to compare different validation methods and store them in an XLS file for further analysis. The following example shows how to do this, using the Mato Grosso data set.

```
# Retrieve the set of samples for the Mato Grosso region (provided by EMBRAPA)
data("samples_mt_4bands")

# create a list to store the results
results <- list()

# adjust the multicores parameters to suit your machine

## SVM model
conf_svm.tb <- sits_kfold_validate(samples_mt_4bands,
                                   folds = 5,
                                   multicores = 2,
                                ml_method = sits_svm(kernel = "radial", cost = 10))

print("== Confusion Matrix = SVM ========================")
```

```
## [1] "== Confusion Matrix = SVM ======================"
```

```
conf_svm.mx <- sits_conf_matrix(conf_svm.tb)
```

```
## Confusion Matrix and Statistics
##
##                Reference
## Prediction      Pasture Soy_Corn Soy_Millet Soy_Cotton Fallow_Cotton
##   Pasture           320        3          6          1             2
##   Soy_Corn            4      336         12          8             0
##   Soy_Millet          3       11        158          0             0
##   Soy_Cotton          4        8          0        338             3
##   Fallow_Cotton       1        0          0          2            23
##   Soy_Sunflower       0        1          1          0             0
##   Cerrado            12        5          3          2             1
##   Forest              0        0          0          0             0
##   Soy_Fallow          0        0          0          1             0
##                Reference
## Prediction      Soy_Sunflower Cerrado Forest Soy_Fallow
##   Pasture                   0       5      2          0
##   Soy_Corn                  7       0      0          0
##   Soy_Millet                2       1      0          0
##   Soy_Cotton                0       0      0          1
##   Fallow_Cotton             0       0      0          0
##   Soy_Sunflower            17       0      0          0
##   Cerrado                   0     372      6          1
##   Forest                    0       1    123          0
##   Soy_Fallow                0       0      0         85
##
## Overall Statistics
##
##  Accuracy : 0.9366
##    95% CI : (0.9246, 0.9471)
##
##     Kappa : 0.9242
##
## Statistics by Class:
##
##                       Class: Pasture Class: Soy_Corn Class: Soy_Millet
## Prod Acc (Sensitivity)         0.9302         0.9231           0.8778
## Specificity                    0.9877         0.9797           0.9901
## User Acc (Pos Pred Value)      0.9440         0.9155           0.9029
## Neg Pred Value                 0.9845         0.9816           0.9872
##                           Class: Soy_Cotton Class: Fallow_Cotton
## Prod Acc (Sensitivity)               0.9602               0.7931
## Specificity                          0.9896               0.9984
## User Acc (Pos Pred Value)            0.9548               0.8846
```

```
## Neg Pred Value                         0.9909                0.9968
##                       Class: Soy_Sunflower Class: Cerrado Class: Forest
## Prod Acc (Sensitivity)                0.6538        0.9815       0.9389
## Specificity                           0.9989        0.9802       0.9994
## User Acc (Pos Pred Value)             0.8947        0.9254       0.9919
## Neg Pred Value                        0.9952        0.9953       0.9955
##                            Class: Soy_Fallow
## Prod Acc (Sensitivity)                0.9770
## Specificity                           0.9994
## User Acc (Pos Pred Value)             0.9884
## Neg Pred Value                        0.9989
```

```
# Give a name to the SVM model
conf_svm.mx$name <- "svm_10"

# store the result
results[[length(results) + 1]] <- conf_svm.mx


# ============== Random Forest ==============================

conf_rfor.tb <- sits_kfold_validate(samples_mt_4bands,
                                    folds = 5,
                                    multicores = 1,
                                    ml_method = sits_rfor(num_trees = 500))
print("== Confusion Matrix = RFOR ======================")
```

```
## [1] "== Confusion Matrix = RFOR ======================"
```

```
conf_rfor.mx <- sits_conf_matrix(conf_rfor.tb)
```

```
## Confusion Matrix and Statistics
##
##                Reference
## Prediction      Pasture Soy_Corn Soy_Millet Soy_Cotton Fallow_Cotton
##   Pasture           339        3          4          0             2
##   Soy_Corn            0      349          5         12             1
##   Soy_Millet          0       10        168          0             0
##   Soy_Cotton          1        2          2        340             2
##   Fallow_Cotton       0        0          0          0            24
##   Soy_Sunflower       0        0          0          0             0
##   Cerrado             4        0          0          0             0
##   Forest              0        0          0          0             0
##   Soy_Fallow          0        0          1          0             0
##                Reference
## Prediction      Soy_Sunflower Cerrado Forest Soy_Fallow
```

```
##   Pasture                    0       0      1          0
##   Soy_Corn                  12       0      0          0
##   Soy_Millet                 1       0      0          2
##   Soy_Cotton                 0       0      0          0
##   Fallow_Cotton              0       0      0          0
##   Soy_Sunflower             13       0      0          0
##   Cerrado                    0     378      1          0
##   Forest                     0       1    129          0
##   Soy_Fallow                 0       0      0         85
##
## Overall Statistics
##
##  Accuracy : 0.9646
##    95% CI : (0.9552, 0.9725)
##
##     Kappa : 0.9577
##
## Statistics by Class:
##
##                       Class: Pasture Class: Soy_Corn Class: Soy_Millet
## Prod Acc (Sensitivity)         0.9855         0.9588           0.9333
## Specificity                    0.9935         0.9804           0.9924
## User Acc (Pos Pred Value)      0.9713         0.9208           0.9282
## Neg Pred Value                 0.9968         0.9901           0.9930
##                           Class: Soy_Cotton Class: Fallow_Cotton
## Prod Acc (Sensitivity)               0.9659               0.8276
## Specificity                          0.9955               1.0000
## User Acc (Pos Pred Value)            0.9798               1.0000
## Neg Pred Value                       0.9922               0.9973
##                       Class: Soy_Sunflower Class: Cerrado Class: Forest
## Prod Acc (Sensitivity)               0.5000        0.9974       0.9847
## Specificity                          1.0000        0.9967       0.9994
## User Acc (Pos Pred Value)            1.0000        0.9869       0.9923
## Neg Pred Value                       0.9931        0.9993       0.9989
##                           Class: Soy_Fallow
## Prod Acc (Sensitivity)               0.9770
## Specificity                          0.9994
## User Acc (Pos Pred Value)            0.9884
## Neg Pred Value                       0.9989
```

```r
# Give a name to the model
conf_rfor.mx$name <- "rfor_500"

# store the results in a list
results[[length(results) + 1]] <- conf_rfor.mx

# choose the output directory
```

```r
WD = getwd()

# Save to an XLS file
sits_to_xlsx(results, file = "./accuracy_mt_ml.xlsx")
```

```
## Saved Excel file ./accuracy_mt_ml.xlsx
```

References

T. Hastie, R. Tibshirani, and Friedman J. *The Elements of Statistical Learning. Data Mining, Inference, and Prediction.* Springer, New York, 2009.