Dynamic Regression Model Performance

Gwen Rino

May 23, 2018

Model Description

To experiment with Dynamic Regression models, I first needed to select the most likely variables for use as the regressor in the xreg argument. I selected five features by looking for the intersection of the most likely variables identified by their `randomForest::importance()` according to a Random Forest model, and those identified by their `earth::evimp()` according to a Multivariate Adaptive Regression Spline (see "dengue/src/FeatureSelection/FeatureSelection.R"). These variables are:

- nonres_guests
- station_max_temp_c
- reanalysis_tdtr_k
- reanalysis_dew_point_temp_k
- reanalysis_specific_humidity_g_per_kg

Experimentation with these five variables revealed that an ARIMA(1,1,1) model with `reanalysis_dew_point_temp_k` as the regressor was the best (see "dengue/src/Models/FinalModel.R"). The model takes the form:

```
# Fit model
model <- auto.arima(ts.final[,"total_cases"],
                    xreg = ts.final[,"reanalysis_dew_point_temp_k"])
summary(model)
```

```
## Series: ts.final[, "total_cases"]
## Regression with ARIMA(1,1,1) errors
##
## Coefficients:
##          ar1      ma1    xreg
##       0.7084  -0.5908  0.7578
## s.e.  0.0961   0.1090  0.5201
##
## sigma^2 estimated as 180.7:  log likelihood=-3754.79
## AIC=7517.59   AICc=7517.63   BIC=7536.95
##
## Training set error measures:
##                         ME     RMSE      MAE MPE MAPE      MASE
## Training set -0.0001752054 13.41438 8.061207 NaN  Inf 0.2206567
##                      ACF1
## Training set 0.0003965629
```

In order to forecast `total_cases` with this model, we need a model for forecasting the xreg variable `reanalysis_dew_point_temp_k`. Experimentation revealed that a fine choice was a Seasonal Naive model (which uses the mean of the value from the season in the past–in this case, the season = the week of the year–as the predicted value for the season in the future). See "dengue/src/Models/FinalModel.R". This Seasonal Naive model for the xreg term is incorporated into the model for forecasting as below:

```
# Model of total_cases with reanalysis_dew_point_temp_k as regressor
model <- auto.arima(ts.final[,"total_cases"],
```

```r
                    xreg  = ts.final[,"reanalysis_dew_point_temp_k"])

# Model of dew point time series
dewpt.model <- snaive(ts.final[,"reanalysis_dew_point_temp_k"])

# Function to forecast using model (returns predictions)

DRfc <- function(h){
  # h is the forecast horizon in weeks
  ptval <- forecast(dewpt.model, h=h)[["mean"]] # predictions for use in xreg
  print(forecast(dengue.model, xreg = rep(ptval))[["mean"]]) # forecast, print results
}
```

Model evaluation

I used the `greybox::ro()` function to cross validate this model using the forecast evaluation on a rolling origin method (500 origins). I forecast at three horizons: 1 week ahead, 6 weeks ahead, and 6 months ahead. The MAE of each of these forecasts show that this model is substantially better than any previous one, and it easily meets the model performance requirements to use it in the dengue prediction application.

```r
# Set up for cross validation
x <- ts.final[,"total_cases"]
xreg <- ts.final[,"reanalysis_dew_point_temp_k"]
ourCall <- "predict(arima(x=data, order=c(1,1,1), xreg=xreg[counti]), n.ahead=h, newxreg=xreg[counto])"
ourValue <- "pred"

## 1 week horizon
returnedValues1 <- ro(x,h=1,origins=500,ourCall,ourValue)
# Calculate MAE
print(paste("1 week horizon MAE = ",
            mean(abs(returnedValues1$actuals[437:936] -
                       returnedValues1$pred[1,]),na.rm = TRUE)))
```

```
## [1] "1 week horizon MAE =  1.06404117223068"
```

```r
## 6 week horizon
returnedValues6 <- ro(x,h=6,origins=500,ourCall,ourValue)
# Calculate MAE
print(paste("6 week horizon MAE = ",
            mean(abs(returnedValues6$actuals[437:936] -
                       returnedValues6$pred[6,]),na.rm = TRUE)))
```

```
## [1] "6 week horizon MAE =  2.99519547461077"
```

```r
## 6 month horizon
returnedValues26 <- ro(x,h=26,origins=500,ourCall,ourValue)
# Calculate MAE
print(paste("6 month horizon MAE = ",
            mean(abs(returnedValues26$actuals[437:936] -
                       returnedValues26$pred[26,]),na.rm = TRUE)))
```

```
## [1] "6 month horizon MAE =  3.35231869865377"
```