Some methods for the quantification of prediction uncertainties for digital soil mapping: Universal kriging prediction variance.

Soil Security Laboratory

2018

1 Universal kriging prediction variance

In the chapter regarding digital soil mapping of continuous variables, universal kriging was explored. This model is ideal from the perspective that both the correlated variables and model residuals are handled simultaneously. This model also automatically generates prediction uncertainty via the kriging variance. It is with this variance estimate that we can define a prediction interval. For this example and the following, a 90% prediction interval will be defined for the mapping purposes. Although for validation, a number of levels of confidence will be defined and subsequently validated in order to assess the performance and sensitivity of the uncertainty estimates.

1.1 Defining the model parameters

First we need to load in all the libraries that are necessary for this section and load in the necessary data.

```
library(ithir)
library(sp)
library(rgdal)
library(raster)
library(gstat)

# Point data
data(HV_subsoilpH)
str(HV_subsoilpH)
```

```
## 'data.frame': 506 obs. of  14 variables:
##  $ X                       : num  340386 340345 340559 340483 340734 ...
##  $ Y                       : num  6368690 6368491 6369168 6368740 6368964 ...
##  $ pH60_100cm              : num  4.47 5.42 6.26 8.03 8.86 ...
##  $ Terrain_Ruggedness_Index: num  1.34 1.42 1.64 1.04 1.27 ...
##  $ AACN                    : num  1.619 0.281 2.301 1.74 3.114 ...
##  $ Landsat_Band1           : int  57 47 59 52 62 53 47 52 53 63 ...
```

```
##  $ Elevation               : num  103.1 103.7 99.9 101.9 99.8 ...
##  $ Hillshading             : num  1.849 1.428 0.934 1.517 1.652 ...
##  $ Light_insolation        : num  1689 1701 1722 1688 1735 ...
##  $ Mid_Slope_Positon       : num  0.876 0.914 0.844 0.848 0.833 ...
##  $ MRVBF                   : num  3.85 3.31 3.66 3.92 3.89 ...
##  $ NDVI                    : num  -0.143 -0.386 -0.197 -0.14 -0.15 ...
##  $ TWI                     : num  17.5 18.2 18.8 18 17.8 ...
##  $ Slope                   : num  1.79 1.42 1.01 1.49 1.83 ...
```

```
# Raster data
data(hunterCovariates_sub)
hunterCovariates_sub
```

```
## class       : RasterStack
## dimensions  : 249, 210, 52290, 11  (nrow, ncol, ncell, nlayers)
## resolution  : 25, 25  (x, y)
## extent      : 338422.3, 343672.3, 6364203, 6370428  (xmin, xmax, ymin, ymax)
## coord. ref. : +proj=utm +zone=56 +south +ellps=WGS84 +datum=WGS84 +units=m +no_defs
```

You will notice for `HV_subsoilpH` that these data have already been intersected with a number of covariates. The `hunterCovariates_sub` are a `rasterStack` of the same covariates (although the spatial extent is smaller).

Now to prepare the data for the universal kriging model.

```
# subset data for modeling
set.seed(123)
training <- sample(nrow(HV_subsoilpH), 0.7 * nrow(HV_subsoilpH))
cDat <- HV_subsoilpH[training, ]
vDat <- HV_subsoilpH[-training, ]
nrow(cDat)
```

```
## [1] 354
```

```
nrow(vDat)
```

```
## [1] 152
```

The `cDat` and `vDat` objects correspond to the model calibration and validation data sets respectively.

Now to prepare the data for the model

```
# coordinates
coordinates(cDat) <- ~X + Y

# remove CRS from grids
crs(hunterCovariates_sub) <- NULL
```

We will firstly use a step wise regression to determine a parsimonious model are the most important covariates.

```
# Full model
lm1 <- lm(pH60_100cm ~ Terrain_Ruggedness_Index + AACN + Landsat_Band1 + Elevation +
```

```
    Hillshading + Light_insolation + Mid_Slope_Positon + MRVBF + NDVI + TWI +
    Slope, data = cDat)

# Parsimous model
lm2 <- step(lm1, direction = "both", trace = 0)
as.formula(lm2)

## pH60_100cm ~ AACN + Landsat_Band1 + Hillshading + Mid_Slope_Positon +
##     MRVBF + NDVI + TWI

summary(lm2)

##
## Call:
## lm(formula = pH60_100cm ~ AACN + Landsat_Band1 + Hillshading +
##     Mid_Slope_Positon + MRVBF + NDVI + TWI, data = cDat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9409 -0.8467 -0.1431  0.6870  3.2195
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        6.55363    1.00729   6.506 2.70e-10 ***
## AACN               0.02652    0.00579   4.580 6.48e-06 ***
## Landsat_Band1     -0.04391    0.01119  -3.925 0.000104 ***
## Hillshading        0.07651    0.02139   3.576 0.000398 ***
## Mid_Slope_Positon  0.88822    0.31849   2.789 0.005582 **
## MRVBF              0.28889    0.09624   3.002 0.002878 **
## NDVI               5.88079    1.06282   5.533 6.22e-08 ***
## TWI                0.11132    0.05657   1.968 0.049889 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.185 on 346 degrees of freedom
## Multiple R-squared:  0.2447,Adjusted R-squared:  0.2294
## F-statistic: 16.01 on 7 and 346 DF,  p-value: < 2.2e-16
```

Now we can construct the universal kriging model using the step wise selected covariates.

```
vgm1 <- variogram(pH60_100cm ~ AACN + Landsat_Band1 + Hillshading + Mid_Slope_Positon +
    MRVBF + NDVI + TWI, cDat, width = 200)
mod <- vgm(psill = var(cDat$pH60_100cm), "Sph", range = 10000, nugget = 0)
model_1 <- fit.variogram(vgm1, mod)

gUK <- gstat(NULL, "hunterpH_UK", pH60_100cm ~ AACN + Landsat_Band1 + Hillshading +
    Mid_Slope_Positon + MRVBF + NDVI + TWI, cDat, model = model_1)
gUK

## data:
## hunterpH_UK : formula = pH60_100cm`~`AACN + Landsat_Band1 + Hillshading +
```

```
## Mid_Slope_Positon + MRVBF + NDVI + TWI ;
## data dim = 354 x 12
## variograms:
##                 model     psill   range
## hunterpH_UK[1]    Nug 0.8895274    0.00
## hunterpH_UK[2]    Sph 0.5204788 1100.54
```

1.2 Spatial mapping

Here we want to produce four maps that will correspond to:

1. The lower end of the 90% prediction interval or 5^{th} percentile.
2. the universal kriging prediction.
3. The upper end of the 90% prediction interval or 95^{th} percentile.
4. The prediction interval range.

For the prediction we use the raster `interpolate` function.

```
UK.P.map <- interpolate(hunterCovariates_sub, gUK, xyOnly = FALSE, index = 1,
    filename = "UK_predMap.tif", format = "GTiff", overwrite = T)
```

Setting the `index` value to 2 lets us map the kriging variance which is needed for the prediction interval. Taking the square root this estimates the standard deviation which we can then multiple for the z value that corresponds to a 90% probability which is `1.644854`. We then both add and subtract that result from the universal kriging prediction to derive the 90% prediction limits.

```
# prediction variance
UK.var.map <- interpolate(hunterCovariates_sub, gUK, xyOnly = FALSE, index = 2,
    filename = "UK_predVarMap.tif", format = "GTiff", overwrite = T)

# standard deviation
f2 <- function(x) (sqrt(x))
UK.stdev.map <- calc(UK.var.map, fun = f2, filename = "UK_predSDMap.tif",
format = "GTiff", progress = "text", overwrite = T)

# Z level
zlev <- qnorm(0.95)
f2 <- function(x) (x * zlev)
UK.mult.map <- calc(UK.stdev.map, fun = f2, filename = "UK_multMap.tif",
 format = "GTiff", progress = "text", overwrite = T)

# Add and subtract mult from prediction
m1 <- stack(UK.P.map, UK.mult.map)

# upper PL
f3 <- function(x) (x[1] + x[2])
UK.upper.map <- calc(m1, fun = f3, filename = "UK_upperMap.tif", format = "GTiff",
```

```
    progress = "text", overwrite = T)

# lower PL
f4 <- function(x) (x[1] - x[2])
UK.lower.map <- calc(m1, fun = f4, filename = "UK_lowerMap.tif", format = "GTiff",
    progress = "text", overwrite = T)
```

Finally to derive the 90% prediction limit range

```
# prediction range
m2 <- stack(UK.upper.map, UK.lower.map)

f5 <- function(x) (x[1] - x[2])
UK.piRange.map <- calc(m2, fun = f5, filename = "UK_piRangeMap.tif", format = "GTiff",
    progress = "text", overwrite = T)
```

So to plot them all together we use the following script. Here we explicitly create a color ramp that follows reasonably closely the pH color ramp. Then we scale each map to the common range for better comparison (Figure 1).

```
# color ramp
phCramp <- c("#d53e4f", "#f46d43", "#fdae61", "#fee08b", "#ffffbf", "#e6f598",
    "#abdda4", "#66c2a5", "#3288bd", "#5e4fa2", "#542788", "#2d004b")
brk <- c(2:14)
par(mfrow = c(2, 2))
plot(UK.lower.map, main = "90% Lower prediction limit", breaks = brk, col = phCramp)
plot(UK.P.map, main = "Prediction", breaks = brk, col = phCramp)
plot(UK.upper.map, main = "90% Upper prediction limit", breaks = brk, col = phCramp)
plot(UK.piRange.map, main = "Prediction limit range", col = terrain.colors(length(seq(0,
    6.5, by = 1)) - 1), axes = FALSE, breaks = seq(0, 6.5, by = 1))
```

1.3 Validating the quantification of uncertainty

One of the ways to assess the performance of the uncertainty quantification is to evaluate the occurrence of times where an observed value is encapsulated by an associated prediction interval. Given a stated level of confidence, we should also expect to find the same percentage of observations encapsulated by its associated prediction interval. We define this percentage as the prediction interval coverage probability (PICP). The PICP was used in both Solomatine and Shrestha (2009) and Malone et al. (2011). To assess the sensitivity of the uncertainty quantification, we define prediction intervals at a number of levels of confidence and then assess the PICP. Ideally, a 1:1 relationship would ensue.

First we apply the universal kriging model `gUK` to the validation data in order to estimate pH and the prediction variance.

```
coordinates(vDat) <- ~X + Y
```

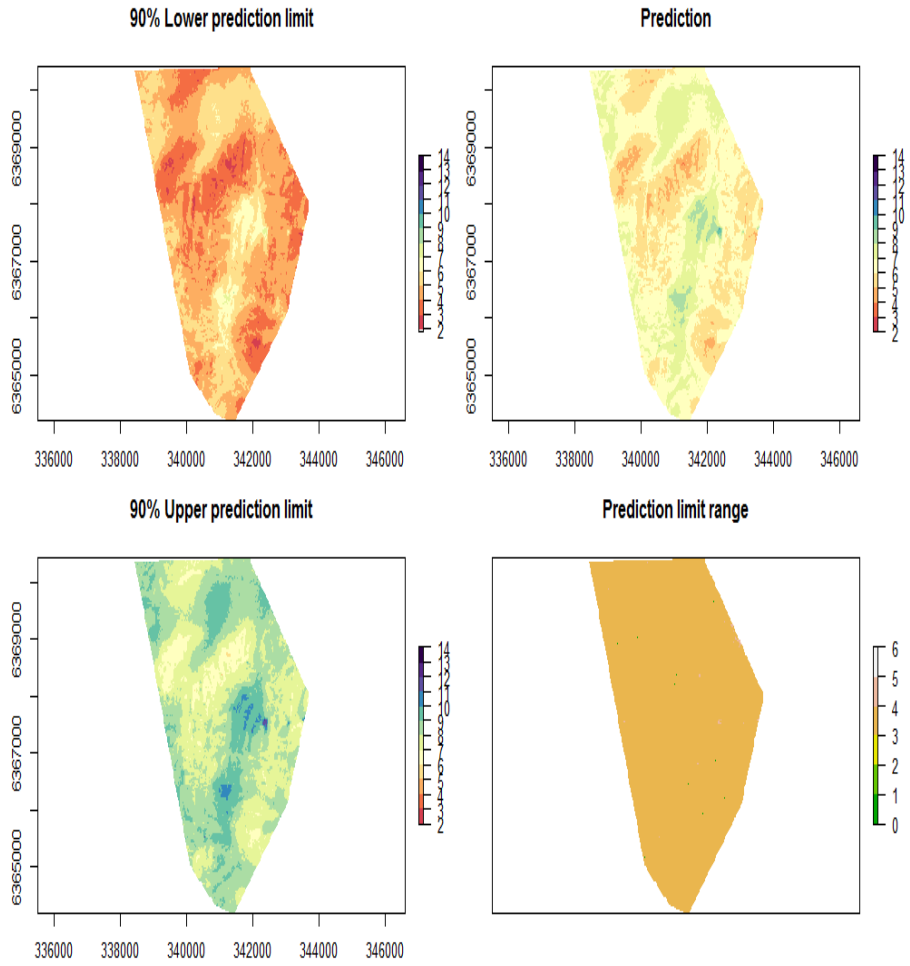

Figure 1: Soil pH predictions and prediction limits derived using a universal kriging model

```
# Prediction
UK.preds.V <- as.data.frame(krige(pH60_100cm ~ AACN + Landsat_Band1 + Hillshading +
    Mid_Slope_Positon + MRVBF + NDVI + TWI, cDat, model = model_1, newdata = vDat))
```

```
## [using universal kriging]
```

```
UK.preds.V$stdev <- sqrt(UK.preds.V$var1.var)
str(UK.preds.V)
```

```
## 'data.frame': 152 obs. of  5 variables:
##  $ X         : num  340559 340780 340861 340905 341131 ...
##  $ Y         : num  6369168 6369166 6368874 6368790 6368945 ...
##  $ var1.pred: num  7.02 7.63 6.59 5.85 5.76 ...
##  $ var1.var : num  1.13 1.18 1.15 1.16 1.12 ...
##  $ stdev     : num  1.06 1.08 1.07 1.07 1.06 ...
```

Then we define a vector of z values for a sequence of probabilities using the `qnorm` function.

```
qp <- qnorm(c(0.995, 0.9875, 0.975, 0.95, 0.9, 0.8, 0.7, 0.6, 0.55, 0.525))
```

Then we estimate the prediction limits for each confidence level.

```
# zfactor multiplication
vMat <- matrix(NA, nrow = nrow(UK.preds.V), ncol = length(qp))
for (i in 1:length(qp)) {
    vMat[, i] <- UK.preds.V$stdev * qp[i]
}

# upper
uMat <- matrix(NA, nrow = nrow(UK.preds.V), ncol = length(qp))
for (i in 1:length(qp)) {
    uMat[, i] <- UK.preds.V$var1.pred + vMat[, i]
}

# lower
lMat <- matrix(NA, nrow = nrow(UK.preds.V), ncol = length(qp))
for (i in 1:length(qp)) {
    lMat[, i] <- UK.preds.V$var1.pred - vMat[, i]
}
```

Then we want to evaluate the PICP for each confidence level.

```
bMat <- matrix(NA, nrow = nrow(UK.preds.V), ncol = length(qp))
for (i in 1:ncol(bMat)) {
    bMat[, i] <- as.numeric(vDat$pH60_100cm <= uMat[, i] & vDat$pH60_100cm >=
        lMat[, i])
}

colSums(bMat)/nrow(bMat)
```

```
##  [1] 0.98026316 0.96052632 0.94078947 0.88157895 0.80921053 0.63815789
##  [7] 0.46052632 0.25000000 0.07236842 0.03947368
```

Plotting the confidence level against the PICP provides a visual means to assess the fidelity about the 1:1 line. As can be seen on Figure 2, the PICP follows closely the 1:1 line.

```
# make plot
cs <- c(99, 97.5, 95, 90, 80, 60, 40, 20, 10, 5)
plot(cs, ((colSums(bMat)/nrow(bMat)) * 100))
```

So to summarize. We may evaluate the performance of the universal kriging model on the basis of the predictions. Using the validation data we would use the `goof` function for that purpose.

```
goof(observed = vDat$pH60_100cm, predicted = UK.preds.V$var1.pred)
```

```
##          R2 concordance      MSE     RMSE      bias
## 1 0.3628938   0.5303005 1.158741 1.076449 0.1709666
```

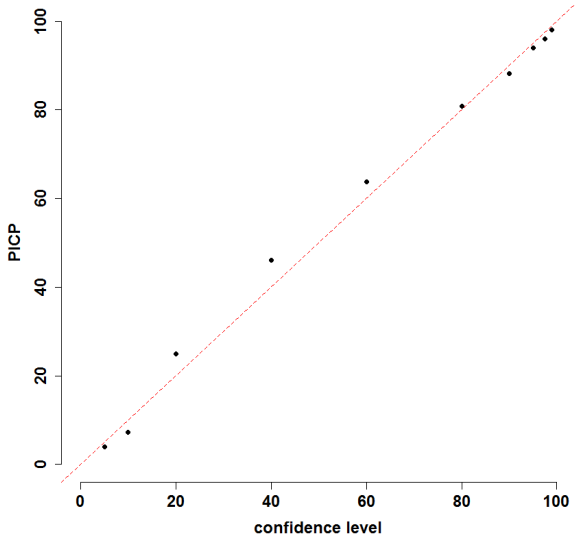

Figure 2: Plot of PICP and confidence level based on validation of universal kriging model.

And then we may assess the uncertainties on the basis of the PICP like shown on Figure 2, together with assessing the quantiles of the distribution of the prediction limit range for a given prediction confidence level (here 90%).

```
cs <- c(99, 97.5, 95, 90, 80, 60, 40, 20, 10, 5)  # confidence level
colnames(lMat) <- cs
colnames(uMat) <- cs
quantile(uMat[, "90"] - lMat[, "90"])
```

```
##       0%      25%      50%      75%     100%
## 3.293101 3.410417 3.461094 3.532295 3.866379
```

As can be noted above, the prediction interval range is relatively homogeneous and this is corroborated on the associated map in Figure 1.

References

Malone, B. P., A. B. McBratney, and B. Minasny
2011. Empirical estimates of uncertainty for mapping continuous depth functions of soil attributes. *Geoderma*, 160:614–626.

Solomatine, D. P. and D. L. Shrestha

2009. A novel method to estimate model uncertainty using machine learning techniques. *Water Resources Research*, 45:Article Number: W00B11.