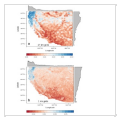

Sep 25, 2019

Protocol for Downscaling Satellite Soil Moisture Estimates using Geomorphometry and Machine Learning

PLOS One

Mario Guevara[1], Rodrigo Vargas[1]

[1]University of Delaware, Department of Plant and Soil Sciences

1 Works for me dx.doi.org/10.17504/protocols.io.6cahase

Mario Guevara
University of Delaware

ABSTRACT

This protocol is composed of five steps for downscaling satellite soil moisture estimates using digital terrain parameters derived from a digital elevation models. We provide an alternative approach to predict soil moisture spatial patterns at higher spatial resolution (compared with current satellite soil moisture estimates) across areas where no information is otherwise available. This approach relies on geomorphometry derived terrain parameters and machine learning models to improve the statistical accuracy and the spatial resolution (from 27km to 1km grids) of satellite soil moisture information. This approach has been tested for this study across the conterminous United States on an annual basis (1991-2016).

- We first derived 15 primary and secondary terrain parameters from a digital elevation model.
- We trained a machine learning algorithm (i.e., kernel weighted nearest neighbors) for each year.
- Terrain parameters were used as predictors and annual satellite soil moisture estimates were used to train the models.
- We validate the models using cross-validation strategies and independent validation.

The explained variance for all models-years was >70% (10-fold cross-validation). The 1km soil moisture grids (compared to the original satellite soil moisture estimates) had higher correlations with field soil moisture observations from the North American Soil Moisture Database (n=668 locations with available data between 1991-2013; 0-5cm depth) than the original product. We conclude that the fusion of geomorphometry methods and satellite soil moisture estimates is useful to increase the spatial resolution and accuracy of satellite-derived soil moisture. This approach can be applied to other satellite-derived soil moisture estimates and regions across the world.

EXTERNAL LINK

https://www.biorxiv.org/content/10.1101/688846v1.abstract

THIS PROTOCOL ACCOMPANIES THE FOLLOWING PUBLICATION

Guevara, M. and Vargas, R.: Downscaling Satellite Soil Moisture using Geomorphometry and Machine Learning, doi:10.1101/688846, 2019.

GUIDELINES

This protocol is used for downscaling satellite soil moisture grids based on terrain parameters derived from digital elevation models (DEMs). The terrain parameters and the satellite soil moisture estimates are required to be in the same projection system.

This protocol is based on an R code that has been tested across the conterminous United States. This protocol can be applied to multiple scales (spatial, temporal) and across specific regions of the world. This protocol assumes that the user has some knowledge of statistics and GIS science. This protocol also assumes that the user has basic R experience performing statistical analysis to multiple geographical data structures (e.g., points, pixels).

To start, install the required software (e.g., R, SAGA-GIS) and its dependencies (e.g., *kknn, raster*), download (or generate) the datasets, and follow the steps of this protocol. Please contact the authors for further information.

MATERIALS

NAME	CATALOG #	VENDOR

NAME	CATALOG #	VENDOR
satellite soil moisture	View	
SAGA GIS for calculating terrain parameters	View	
digital elevation model	View	
R for statistical computing	View	
reference preprint	View	

MATERIALS TEXT

Download the same data used in the publication here:
Guevara, M., R. Vargas (2019). Annual soil moisture predictions across conterminous United States using remote sensing and terrain analysis across 1 km grids (1991-2016), HydroShare,https://doi.org/10.4211/hs.b8f6eae9d89241cf8b5904033460af61

SAFETY WARNINGS

Read carefully every potential warning or error message and ask R about the proper arguments in the function generating that warning or error (e.g., if the error is generated when using the function *list.files*, then type in R *?list.files,* and verify that you are using the correct function arguments). Be sure all inputs are in the same geographical projection system.

Be sure that the selected area of interest is large enough to include not less that 50 satellite soil moisture pixels. This is an arbitrary minimum number of pixels in which the cross validation strategy will perform with no issues most of times (e.g., 95% of times). Be sure that you have a good quality DEM for the area of interest that you can use in SAGA-GIS for generating terrain parameters.

BEFORE STARTING

Be sure that R (the libraries *kknn, rgdal* and *raster*) are properly installed. Be sure that SAGA-GIS is installed and use the Module Basic (or Standard) Terrain Analysis for deriving (automatically) terrain parameters from the digital elevation model of your study area https://sagatutorials.wordpress.com/basic-terrain-analysis/. Save these terrain parameters in your local computer.

We provide terrain parameters at 1km pixel size across the conterminous United States here: https://doi.org/10.4211/hs.b8f6eae9d89241cf8b5904033460af61, This site also include the satellite soil moisture yearly means (1991-2016) for the same area. Thus, before start please download satellite soil moisture (coarse scale training data) for this or for your area of interest, and download (or use SAGA-GIS to derive) the aforementioned terrain parameters (the fine scale prediction factors).

1

PREPARE PREDICTION FACTORS: This first stetp is to prepare the prediction factors. Once R is installed and running, please load the library *raster* and import the DEM terrain parameters. This example is using a geographical coordinate system (lat, long) in degrees ('+proj=longlat +datum=WGS84 +no_defs'). We will consider (as in this case) that the topographic prediction factors are in a specified path in the local computer (e.g., 'path/to/topographic/files') and that all of them (n=15) have a SAGA-GIS native *.sdat format (these can be *.tif, *.asc or any other generic raster files). If you have your own set of prediction factors in the same projection system and pixel size, reproject (if needed) to a geographical coordinate system (because that is the projection of the ESA-CCI satellite soil moisture product that we are using for testing this protocol), convert to a *'SpatialPixelDataFrame'* object and go to step 2.

```
##library to manage raster files
library(raster)
##path to prediction factors (e.g., a folder with *.tif predictors)
lis1 <- 'path/to/topographic/files'
##object with the names of the predictor raster files
#(can be *.tif or *.asc, or *.img files)
lis2 <-list.files(lis1, full.name=TRUE, pattern ='sdat')
##make a raster stack of the files
x <- stack(lis2)
#function to remove NAs
NA2mean <- function(x) replace(x, is.na(x), median(x, na.rm = TRUE))
#convert to spatial pixels dataframe (this will take overnight on conventional systems, but will
work)
x=as(x,'SpatialPixelsDataFrame')
#for global downscaling use this line
#x=as(s,'SpatialPixelsDataFrame')
#transform NAs to median values of layer
x@data[] <- lapply(x@data, NA2mean)
```

The resulting object *x* (from this step) contains the prediction factors used as the basis of the downscaling process. This object containing all terrain parameters is an internal R format tyme known as '*SpatialPixelsDataFrame*'.

2 PREPARE RESPONSE VARIABLE: The next step is about getting and importing the soil moisture dataset that is going to be downscaled to a finer pixel size. For our paper we used the European Space Agency-Climate Change Initiative soil moisture product (ESA-CCI). These data sets are provided here: https://www.esa-soilmoisture-cci.org/ . These data is provided yearly folders and in *.nc files (one file per day) and we can import these files in R using the following code for listing the folders with the satellite soil moisture data. Download the satellite soil moisture maps of your convenience and place them in your working directory.

```
#list.dirs is a function used for generating a list of the annual soil moisture #folders as
provided by the ESA-CCI initiative
dirs <- list.dirs()[-1]
#we can start a loop but first test with a yearly mean or median first
#test with i=1 before the loop
#i=1
#for (i in 1:26){#if you want it to make it daily use 1:366,
#1:26 means 26 years between 1991 and 2016
#stack soil moisture records for a given folder/year
#in this case i = 1 = year 1991
#change the nc extention if your satellite data is in another format
r <- stack(paste0(dirs[i], '/',list.files(dirs[i], pattern='nc')), varname='sm')
#crop, if needed, to area of interes using an extent
#this is the extent of the prediction factors
#consider that satellite soil moisture has 27km pixels, select at least 30 - 50 of #this pixels to
build the models. Higher pixels for training models higher accuracy.
r <- crop (r, extent(x))
#calculate median year value from daily estimates,
#avoid this line if you want to go in day or month basis
r <- calc(r, median,  na.rm=TRUE)
```

The resulting raster object *r* will be harmonized with the result of step 1 in order to build a regression matrix for soil moisture. In this case *r* is the mean value of soil moisture estimates avaialable in the ESA-CCI product for the year 1991.

3 REGRESSION MATRIX: Next step is building a regresion matrix. This is a data frame with columns for coordinates, colum for soil moisture data and columns of terrain parameters. This regression matrix will be used for building a model in the next step.

```
#define lat long projection
proj4string(r) <- CRS("+proj=longlat +datum=WGS84 +no_defs")
#convert to data frame
df=as.data.frame(r, xy=T)
#remove no data values
df=na.omit(df)
#define column coordinates
coordinates(df)=~x+y
#and lat long projection system
proj4string(df) <- CRS("+proj=longlat +datum=WGS84 +no_defs")
#overlay soil moisture centroids and prediction factors (x)
ov=over(df, x)
#generate a data frame
d=as.data.frame(df)
#combine extracted values
y=cbind(d,  ov)
#remove no data values
y=na.omit(y)
#training set year i
z=data.frame(z=y[,3],y[,1:2],  y[,4:18])
#print dimensions of the regression matrix
print(dim(z))
ion (correlation obs vs mod [cd] and RMSE as well as N) in a data frame named results
#empty dataframe to store results
results<-data.frame(year=numeric(), cor=numeric(),rmse=numeric(),
n=numeric(),kernel=as.character(),stringsAsFactors=FALSE,k=numeric()
```

The resulting object is a data frame *z*, with the coordinate colums of centroids of ESA-CCI pixels and the corresponding values for those coordinates of the terrain parameters. This regression is used for building models and generating digital soil moisture maps. We additionally generate an empty data frame *results* for saving the modeling results (cross validation accuracy).

4 MODELING SOIL MOISTURE: Next we build a prediction model. We used as in our paper a kernel based machine learning model, but it can be any model able to account for non linear relationships see methos section of our paper. We start loading the required library *kknn.* The 'optimal' model parameters k and the kernel function will be stored in the previously generated *results* object. In this results object we will also save the accuracy indicators (cor and rmse) indicating the correlation between observed and predicted and the corresponding root mean squared error. This accuracy metrics are derived from a 10-fold cross validation strategy.

```
library(kknn)
#you must select the best parameters by tunning them with CV,  the parameter K and the parameter
kernel
##
#k between 1 and 50
kmax=50
#find optimal k and kernel type via 10 fold cross validation
knnTuning <- train.kknn(z~., data=z, kmax = kmax, distance = 2,
kernel = c("rectangular", "triangular", "epanechnikov","gaussian", "rank", "optimal"),
          ykernel = NULL, scale = TRUE,kcv=10)
#identify best kernel
n<-which(knnTuning$best.parameters$kernel==c("rectangular", "triangular",
"epanechnikov","gaussian", "rank", "optimal"))
#store best parameters
mejoresresultados <- data.matrix(unlist(knnTuning$fitted.values[[(kmax*(n-
1))+knnTuning$best.parameters$k]]))
#lowest rmse
rmse <- sqrt(knnTuning$MEAN.SQU[knnTuning$best.parameters$k,n])
#highest correlation
cd <- cor(z[,1], mejoresresultados)
#modeling year (1991-2016)
year=1990+i
#prepare best kernel
as.character(unlist(knnTuning$best.parameters[1]))
#prepare best k
k=as.numeric(knnTuning$best.parameters[2])
#store results for year i in the data frame
results[i,1]<-year
results[i,2]<-cd
results[i,3]<-rmse
results[i,4]<-dim(z)[1]
results[i,5]<-unlist(knnTuning$best.parameters[1])
results[i,6]<-k
#print accuracy results on screen
print(results)
```

The resulting object of this section is a model that we can use to generate predictions. This model is cross validated and the accuracy report is in the object *results*. In this example we predict across all CONUS using 1km grids.

5 PREDICTION ACROSS ALL AREA OF INTEREST: This is the final step. We apply the coefficients of the model (for each day or year) to all the prediction domain (defined by the terrain parameters). Thus we obtain a soil prediction for each pixel across the study area. These prediction values are acoupled with the coordinates of the center of each pixel and a digital soil moisture map is generated.

```
##build a model with best parameters, predict to all CONUS and make a map of it
mejorKKNN <- kknn(z~.,train=z,test=x,kernel=unlist(knnTuning$best.parameters[1]),
                        scale=TRUE,k=as.numeric(knnTuning$best.parameters[2]))
#add fitted values to the prediction domain
x$kknn=mejorKKNN$fitted.values
#make a raster of the prediction
r <- raster(x['kknn'])
#visualize the map
#plot(r)
#proj4string(r) <- CRS("+proj=longlat +datum=WGS84 +no_defs")
#save results
writeRaster(r, file=paste0('predicted-', 1990+i, '-esa-sm-topo-CONUS.tif'), overwrite=TRUE) ###
change the name of the output files if necessary
#print accuracy results on screen
print(results)
#save accuracy results
write.csv(results, file=paste0('accuracy-', 1990+i, '-esa-sm-topo-CONUS.csv'))
```

The resulting objects from this step are a data frame *results* with the accuracy report of the model for each year, and the object *r*, a raster file containing the predictions of soil moisture across the study area. These objects are going to be saved in the working directory as a *.tif file for the raster object and in a *.csv for the *results* file.