FSVM: Predictions

Idaho Department of Fish and Game

Robert Ritson, WMI Research Associate

2021-06-24

```
## Installing package into 'C:/Users/rritson/Documents/R/win-library/4.0'
## (as 'lib' is unspecified)
```

Getting Predictions from Trained `fsvm` models

Once you have trained machine learning models on field data with `fsvm`, the next step is to get useful predictions from the models for the areas within Idaho which you are interested. These can be anything from an animal home range, migration corridor, game management unit, or even an entire region. However, the larger the extent, the longer the amount of time required to calculate the prediction will take. For the purposes of this vignette, we will get fine scale vegetation predictions with one ungulate home range.

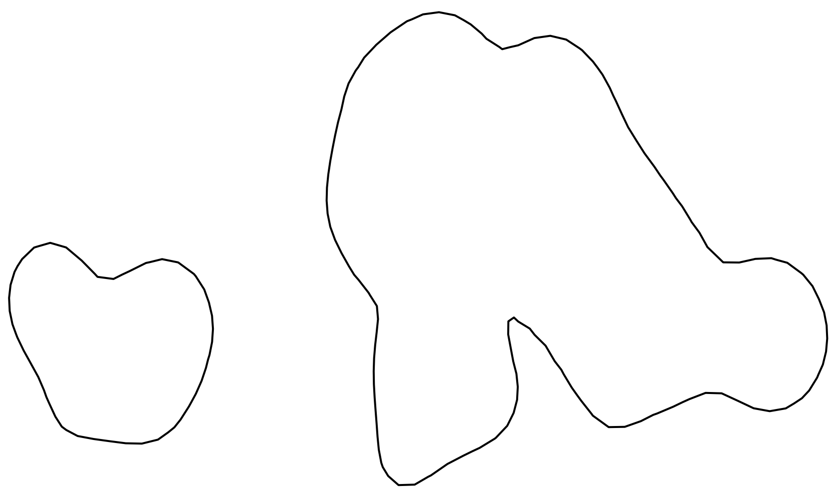
Step 1: Extract QuadPoly_IDs

The shapefile we will be using is an ungulate VHF home range contour available on the IDFG "KEEP" drive. Once loaded into R, it needs to be projected to Idaho Transverse Mercator in order to extract the QuadPoly_IDs from the eCognition polygons (see 'fsvm_workflow_vignette.pdf' for details).

```
#Load packages
require(fsvm)
require(rgdal)
require(sf)
require(sp)

#Load shapefile
filepath <- "K:/Wildlife/Wildlife Research/Vegetation Sampling/Vegetation sampling/
             Summer home ranges/VHF Summer Home Ranges/contours"
homerange_shp <- rgdal::readOGR(dsn = filepath, layer = "39")

#Set projection
sp::proj4string(homerange_shp) <- sp::CRS("+proj=longlat +ellps=WGS84
                                          +datum=WGS84 +no_defs")

#Transform to Idaho Transverse Mercator projection
```

```
idtm <- sp::CRS("+proj=tmerc +lat_0=42 +lon_0=-114 +k=0.9996 +x_0=2500000 +y_0=1200000
                +datum=NAD83 +units=m +no_defs +ellps=GRS80 +towgs84=0,0,0")
homerange_shp <- sp::spTransform(homerange_shp, idtm)
```

```
#Plot and save homerange kernal
plot(homerange_shp)
```



```
rgdal::writeOGR(homerange_shp,
      dsn = "A:/Fine scale vegetation analysis/fsvm_package/Vignette_Examples",
      layer = "HR39_idtm",
      driver = "ESRI Shapefile",overwrite_layer = T)
```

After we have saved the re-projected homerange kernal, we are ready to move on to extracting the QuadPoly_IDs.

Extract QuadPoly_IDs within homerange

In order to speed up the extraction process, we are going to restrict the number of 100k Quads the Python script has to load and loop through by only selecting those which intersect with our home range shapefile.

```
#Speed up extraction by restricting 100k quads
##Select 100k quads intersecting with homerange
quads<-rgdal::readOGR(paste(system.file('esri',package = 'fsvm',mustWork = T),
                            '100kquads',sep = '/'),layer="quad100k_proj")
sf.quads <- sf::st_as_sf(quads)
homerange.sf <- sf::st_as_sf(homerange_shp)
X <- sf::st_intersection(homerange.sf,sf.quads)
```

```r
uid <- X$UID #use this list in 'fsvm::set_extent_manual'
rm(quads,sf.quads,X,homerange.sf)

quad.sel <- fsvm::set_extent_manual(extent = "UID_100k", selections = uid)
quad.sel
rm(uid)
```

Instead of loading and looping through all 75 100k USGS Quads, the Python script can now be told to only loop through the 4 which the home range shapefile intersects. Now we are ready to execute the Python script to get the QuadPoly_IDs within the home range.

```r
#Extract QuadPoly_IDs
hr.path<-"A:/Fine scale vegetation analysis/fsvm_package/Vignette_Examples/HR39_idtm.shp"
#extent.qpid <- fsvm::py_extract_quadpolyID(py.path = "C:/Python27/ArcGIS10.6/python.exe",
#     fielddata.path = hr.path,
#     output.gdb.path = "A:/Fine scale vegetation analysis/fsvm_package/Vignette_Examples",
#     output.folder.path = "A:/Fine scale vegetation analysis/fsvm_package/Vignette_Examples/Quads_Outp
#     output.dbf = "HR39_quads.dbf",
#     output.RData = "None",
#     intercept.feature = "None",
#     newgdb.name = "Output.gdb",
#     quad.sel = quad.sel,
#     export = T)
#head(extent.qpid)
```

Step 2: Get Prediction Covariates

Now that we know which QuadPoly_IDs we are interested in, next we can get the covariate data corresponding to those polygons to feed into the trained vegetation models.

```r
#Extract prediction covariates
#predcovs <- fsvm::getPredCovs(extent = extent.qpid, export = F)
#head(predcovs)
```

Step 3: Get Model Predictions

Get predictions for a single vegetation species

Now that we have the covariate data set, we can use the best selected trained model to predict the distribution of a particular vegetation species. Let's try looking at the probability of sagebrush (Artemisia tridentata) presence within the home range.

```r
#Get predictions from a trained 'fsvm' vegetation species model
#HR39_Sagebrush_pres <- fsvm::fsvm_predict(pred_covs = predcovs,
#                                          model = "Artemisia tridentata.RData",
#                                          folder = "G2",
#                                          type = "presence")
#head(HR39_Sagebrush_pres)
```

Not exactly eye-catching - Below we can map out these numbers to visualize the predictions with `getCentroidMap`.

{forthcoming...}

Get predictions for multiple vegetation species

Get predictions for all trained vegetation models. . . .

```
#Predict from models
#fsvm::getPredictions()
```

Step 4: Map Predictions

Create a map visualizing model predictions within home range extent. . .

```
#Map predictions
#fsvm::getCentroidMap()

#Plot prediction map
#plot(output_map)
```