Step 6.1 Identifying error sources at plot scale

6.1.2 Analyzing link between misclassifications and variables

Diane ESPEL

2025-06-20

Contents

1 Objectives **1**

2 Script explanation **1**

2.1 **Clean environment and graphics** 1

2.2 **Load required packages** 2

2.3 **Define Global Variables** 2

2.4 **Set working directory** 2

2.5 **Load learning data informations** 3

2.6 **Analyse data with FAMD** 3

1 Objectives

The main objective of this script is to perform a **Factor Analysis of Mixed Data (FAMD) for each level of habitat classification** in order to better understand the relationships between prediction errors and the spectral predictor variables. FAMD is particularly suited for datasets that include both quantitative and qualitative variables, which is typical in ecological modeling that combines spectral and categorical field data.

This analysis helps to:

- Assess which variables are most associated with prediction errors.
- Explore the multivariate structure of predictor variables.
- Visualize relationships between individuals (plots) and their classification accuracy.

2 Script explanation

2.1 Clean environment and graphics

```r
rm(list = ls())  # Clear all objects from the R environment to start fresh
graphics.off()  # Close all graphics devices (if any plots are open)
```

2.2 Load required packages

```r
library(FactoMineR)  # Tools for multivariate data analysis, including PCA, MCA, FAMD, etc.
library(factoextra)  # Visualization tools for the results of multivariate data analyses from
↪  FactoMineR
library(viridis)  # Color palettes optimized for perceptual uniformity and colorblind-friendliness
library(dplyr)  # Grammar of data manipulation (filter, select, mutate, summarize, etc.)
```

2.3 Define Global Variables

Note: A global variable is a variable defined outside of any function. This means the variable is accessible from any part of the code, including inside functions. A global variable retains its value throughout the execution of the R script unless it is explicitly modified in the code.

It is important to define at a minimum:

- the `"District"`: the archipelago of interest (e.g. "CRO" for Crozet archipelago)
- the `"Island"`: the island within the archipelago of interest (e.g. "POS" for Possession island)
- the `"Satellite1"`: the satellite name for multispectral imagery
- the `"maxTypoLevel"`: the maximum typology level

```r
District = "CRO"  # 3-letter code for archipelago (e.g. Crozet)
Island = "POS"  # 3-letter code for island (e.g. Possession)
Satellite1 = "Pleiades"  # satellite name of multispectral imagery
maxTypoLevel = 4  # Define maximum typology level
```

2.4 Set working directory

You must define a general root directory (`"localscratch"`) that serves as the base path for your input and output data. This directory should point to the local environment where:

- input learning data informations are located under `"data/Learning_data/NewTypo"`
- input results from final model are located under `"results/Model/Final_model"`
- outputs results will be saved under `"results/Model/Final_model"`

```r
# Base local path (customize to your local environment)
localscratch = paste0("/scratch/despel/CARTOVEGE/")
# localscratch = paste0('your_local_path/')

# Path to open learning data
open_learning_new_path = paste0(localscratch, "data/Learning_data/NewTypo")

# Path to open results from final model
open_final_model_path = paste0(localscratch, "results/Model/Final_model")

# Path to save results
save_final_model_path = paste0(localscratch, "results/Model/Final_model")
```

2.5 Load learning data informations

```
# Open learning data
FILE1 = paste0(open_learning_new_path, "/", "Learning_plots_", District, "_", Island,
    "_", Satellite1, "_", Year1, "_ALL_SOURCES_EPSG32739.csv")
learning_data <- read.csv(FILE1, sep = ";", dec = ".", stringsAsFactors = FALSE)  #
 ↪  `stringsAsFactors=F` ensures character strings don't import as factors

# Remove useless columns to keep only variable data
variable_data <- learning_data %>%
    select(-matches(c("^Hab_L[1-4]$", "^Hab_L[1-4]_corr$", "xcoord_m", "ycoord_m")))
```

2.6 Analyse data with FAMD

The script loops over each classification level (from 1 to maxTypoLevel), allowing a stratified analysis of error patterns across increasing thematic complexity.

For each level:

- The script reads a CSV file (`Local_errors_Final_RF_...csv`) that contains observed vs. predicted labels and error flags (`YES`/`NO`) at the plot level.
- It performs a left join (`left_join()`) between this error dataset (`error_data`) and the original predictor dataset (`variable_data`) based on a common identifier (`ID`), effectively enriching the dataset with explanatory variables.
- It removes unnecessary metadata columns (e.g., coordinates, predictions) and retains only predictor variables and the `Error` column for analysis with FAMD.
- FAMD is conducted on the cleaned dataset using FactoMineR::FAMD. This generates:
  - Principal components (axes)
  - Variable contributions
  - Representation qualities ($\cos^2$ values)

  Only the first 5 dimensions are retained, as specified by `ncp = 5`.
- A scree plot of eigenvalues is generated to visualize the importance of each axis.
- The quality of representation and contribution of all variables to the FAMD dimensions are saved in separate CSV files and additional barplots show the contribution of each variable to the first two dimensions individually.
- A global factor map for individuals is created, colored by $\cos^2$ to reflect the quality of representation.
- A final version colors individuals by their habitat class, using a viridis color scale for visual clarity. This allows inspection of how misclassifications relate to habitat types.

```
# Loop through each habitat classification level
for (l in seq (1:maxTypoLevel)){

  print(paste0("Processing classification level ", l))

  # Open Local error to join errors to spectral _data

 ↪  FILE2=paste0(open_final_model_path,"/","Local_errors_Final_RF_",type_model,"_model_",District,"_",Island,"_",Sa
```

```r
error_data <- read.csv(FILE2, sep=";",dec=".",stringsAsFactors=FALSE) # `stringsAsFactors=F` ensures character strings don't import as factors

# left join between predictors and prediction errors at plot scale
error_with_variable_data=left_join(error_data,variable_data,by="ID")


# Define the folder for each level of classification
LevelFolder=paste0(save_final_model_path,"/","Hab_L",l)
dir.create(LevelFolder,showWarnings=F) # ShowWarnings=F to remove warnings message if file already exists


# Select columns of interest : Error (Yes/No) and variables only
famd_df<- error_with_variable_data %>% select(-matches(c("ID","xcoord_m","ycoord_m","Yobs","Ypred")))
set.seed(123)


# Apply FAMD
res_famd=FAMD(famd_df,  #dateframe with n rows and p columns
              ncp = 5, # maximum number of dimensions
              graph = F)


# Variances  ------------------------------------------------------------

# Get eigen values
eig.val=as.data.frame(res_famd$eig)
FILE2=paste0(LevelFolder,"/","FAMD_",District,"_",Island,"_",Satellite1, "_", Year1, "_ALL_SOURCES_L",l,"_Eigenvalues.csv")
write.table(eig.val,FILE2, sep = ";", dec = ".", row.names = FALSE)

# Plot of Eigen values
NOMpng=paste0(LevelFolder,"/","FAMD_",District,"_",Island,"_",Satellite1, "_", Year1, "_ALL_SOURCES_L",l,"_Eigenvalues.png")
png(file = NOMpng, width = 500, height = 400)
p=fviz_eig(res_famd,title="Eigen values")+
  theme(text = element_text(size = 17),
        axis.title = element_text(size = 17),
        axis.text = element_text(size = 15))
print(p)
dev.off()



# Graphs for all variables  ---------------------------------------------------------

# Get results for variables
var=get_famd_var(res_famd) # quantitative and qualitative variables

# Quality of the representation
Quality=as.data.frame(var$cos2)
FILE3=paste0(LevelFolder,"/","FAMD_",District,"_",Island,"_",Satellite1, "_", Year1, "_ALL_SOURCES_L",l,"_AllVariables_quality_of_representation.csv")
write.table(Quality,FILE3, sep = ";", dec = ".", row.names = FALSE)

# All variables Contributions to dimensions
Contrib=as.data.frame(var$contrib)
```

```r
FILE4=paste0(LevelFolder,"/","FAMD_",District,"_",Island,"_",Satellite1, "_", Year1,
↪  "_ALL_SOURCES_L",l,"_AllVariables_contributions_to_axis.csv")
write.table(Contrib,FILE4, sep = ";", dec = ".", row.names = FALSE)


# Variable graphs
NOMpng=paste0(LevelFolder,"/","FAMD_",District,"_",Island,"_",Satellite1, "_", Year1,
↪  "_ALL_SOURCES_L",l,"_AllVariables.png")
png(file = NOMpng, width = 700, height = 700)
p=fviz_famd_var(res_famd, repel = T, #repel to unlabel data points with overlaps
                labelsize=3,pointsize=4)+
  ylim (0,1)+
  xlim(0,1)+
  theme(text = element_text(size = 17),
        axis.title = element_text(size = 17),
        axis.text = element_text(size = 17))
print(p)
dev.off()

# All Variables Contributions to dimensions
for (dim in 1:2) {
  png(file.path(LevelFolder, paste0("FAMD_", District, "_", Island, "_", Year1, "_", Type, "_L",
   ↪  l, "_Variables_contributions_DIM", dim, "_FAMD.png")), width = 500, height = 400)
  p=fviz_contrib(res_famd, "var", axes = dim) +
    theme(text = element_text(size = 17), axis.title = element_text(size = 17), axis.text =
     ↪  element_text(size = 10))
  print(p)
  dev.off()
}


# Graphs of quantitative variables -----------------------------------------------------------

# Get results for quantitative variables
quanti.var <- get_famd_var(res_famd, "quanti.var") # for quantitative variables

Quality=as.data.frame(quanti.var$cos2)
FILE5=paste0(LevelFolder,"/","FAMD_",District,"_",Island,"_",Satellite1, "_", Year1,
↪  "_ALL_SOURCES_L",l,"_Quantitative_Variables_quality_of_representation.csv")
write.table(Quality,FILE5, sep = ";", dec = ".", row.names = FALSE)

Contrib=as.data.frame(quanti.var$contrib)
FILE6=paste0(LevelFolder,"/","FAMD_",District,"_",Island,"_",Satellite1, "_", Year1,
↪  "_ALL_SOURCES_L",l,"_Quantitative_Variables_contributions_to_axis.csv")
write.table(Contrib,FILE6, sep = ";", dec = ".", row.names = FALSE)


# Graphs
NOMpng=paste0(LevelFolder,"/","FAMD_",District,"_",Island,"_",Satellite1, "_", Year1,
↪  "_ALL_SOURCES_L",l,"_Quantitative_Variables.png")
png(file = NOMpng, width = 700, height = 700)
p=fviz_famd_var(res_famd, "quanti.var", repel=T, col.var = "contrib",axes=c(1,2),gradient.cols =
↪  c("#00AFBB", "#E7B800", "#FC4E07"),labelsize=3,pointsize=4)+
  theme(text = element_text(size = 17),
        axis.title = element_text(size = 17),
        axis.text = element_text(size = 15))
print(p)
dev.off()
```

```
# Graphs of qualitative variables ----------------------------------------------------------

# Get results for qualitative variables
quali.var <- get_famd_var(res_famd, "quali.var") # for qualitative variables

Quality=as.data.frame(quali.var$cos)
FILE7=paste0(LevelFolder,"/","FAMD_",District,"_",Island,"_",Satellite1, "_", Year1,
↪  "_ALL_SOURCES_L",l,"_Qualitative_Variables_quality_of_representation.csv")
write.table(Quality,FILE7, sep = ";", dec = ".", row.names = FALSE)

Contrib=as.data.frame(quali.var$contrib) # variables Contributions to dimensions
FILE8=paste0(LevelFolder,"/","FAMD_",District,"_",Island,"_",Satellite1, "_", Year1,
↪  "_ALL_SOURCES_L",l,"_Qualitative_Variables_contributions_to_axis.csv")
write.table(Contrib,FILE8, sep = ";", dec = ".", row.names = FALSE)

#Graphs
NOMpng=paste0(LevelFolder,"/","FAMD_",District,"_",Island,"_",Satellite1, "_", Year1,
↪  "_ALL_SOURCES_L",l,"_Qualitative_Variables.png")
png(file = NOMpng, width = 900, height = 700)
p=fviz_famd_var(res_famd, "quali.var", repel=T,col.var = "contrib", gradient.cols = c("#00AFBB",
↪  "#E7B800", "#FC4E07"),labelsize=4,pointsize=0.9)+
  ylim (-7.5,7.5)  + xlim(-10,10) +
  theme(text = element_text(size = 17),
        axis.title = element_text(size = 17),
        axis.text = element_text(size = 15))
print(p)
dev.off()

NOMpng=paste0(LevelFolder,"/","FAMD_",District,"_",Island,"_",Satellite1, "_", Year1,
↪  "_ALL_SOURCES_L",l,"_Qualitative_Variables2.png")
png(file = NOMpng, width = 700, height = 700)
p=plot(res_famd, choix="quali",cex=1,cex.axis=1,font.axis=1.5,ylim=c(-7.5,7.5),xlim=c(-10,10))
print(p)
dev.off()

NOMsvg=paste0(LevelFolder,"/","FAMD_",District,"_",Island,"_",Satellite1, "_", Year1,
↪  "_ALL_SOURCES_L",l,"_Qualitative_Variables2.svg")
svg(file = NOMsvg)
p=plot(res_famd, choix="quali",cex=1,cex.axis=1,font.axis=1.5,ylim=c(-7.5,7.5),xlim=c(-10,10))
print(p)
dev.off()



# Graphs of individuals----------------------------------------------------------


# Individual graphs cos2
NOMpng=paste0(LevelFolder,"/","FAMD_",District,"_",Island,"_",Satellite1, "_", Year1,
↪  "_ALL_SOURCES_L",l,"_Individuals.png")
png(file = NOMpng, width = 700, height = 700)
p=fviz_famd_ind(res_famd, repel=TRUE, col.ind = "cos2", gradient.cols = c("#00AFBB", "#E7B800",
↪  "#FC4E07"),labelsize=5,pointsize=4,ggrepel = TRUE, geom = c("text","point"))+
  ylim (-7.5, 7.5)  + xlim(-10,10) +
  theme(text = element_text(size = 17),
        axis.title = element_text(size = 17),
        axis.text = element_text(size = 15))
print(p)
```

```
  dev.off()


  NOMpng=paste0(LevelFolder,"/","FAMD_",District,"_",Island,"_",Satellite1, "_", Year1,
↪  "_ALL_SOURCES_L",l,"_Individuals2.png")
  png(file = NOMpng, width = 700, height = 700)
  p=plot(res_famd, choix = "ind",ylim=c(-7.5,7.5),xlim=c(-10,10))
  print(p)
  dev.off()


  # Color individuals according to class
  NOMpng=paste0(LevelFolder,"/","FAMD_",District,"_",Island,"_",Satellite1, "_", Year1,
↪  "_ALL_SOURCES_L",l,"_Individuals_Type_habitat.png")
  png(file = NOMpng, width = 700, height = 700)
  p=fviz_famd_ind(res_famd, habillage=famd_df[,1],label="none",
                  geom=c("point"),pointsize=3,
                  geom.ind=c("point"),
                  show.legend=F)+
    scale_y_continuous(limits = c(-7.5,7.5)) + # Limiter l'axe des y à -12 et 12.5
    scale_x_continuous(limits = c(-10, 10)) + # Limiter l'axe des X à -12 et 20
    scale_color_viridis(discrete = TRUE,begin=0.1,end=1)+
    scale_fill_viridis(discrete = TRUE,begin=0.1,end=1)+
    theme_classic() +
    theme(text = element_text(size = 17),
          axis.title = element_text(size = 17),
          axis.text = element_text(size = 15),
          plot.title = element_text(size = 22, face = "bold"),
          legend.position = "right",
          legend.text = element_text(size = 14),
          legend.title = element_text(size = 16, face = "bold")) +
    labs(title = paste("Individuals - FAMD - Niveau ", l),
         fill = "Types d'habitat")

  print(p)
  dev.off()


  # Save the entire current R workspace (objects, variables, etc.) to a .RData file named according
↪  to the classification level
  save.image(paste0(LevelFolder,"/","FAMD_level_",l,".RData"))



}
```

This script is a comprehensive diagnostic tool for classification errors in habitat mapping using remote sensing. By linking prediction errors with spectral, terrain and categorical data via FAMD, it helps identify which variables or classes contribute most to misclassifications, providing insights to refine models or field validation strategies.