Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *doi: 10.18637/jss.v000.i00*

Bayesian inference for post-processing of remote sensing image classification

Gilberto Camara
Nat Inst for Space Research
Brazil

Renato Assunção
ESRI Inc. and UFMG
USA and Brazil

Rolf Simoes
Nat Inst for Space Research
Brazil

Alexandre Carvalho
Inst Applied Economic Research
Brazil

Felipe Souza
Nat Inst for Space Research
Brazil

Pedro R. Andrade
Nat Inst for Space Research
Brazil

Abstract

The abstract of the article.

Keywords: Bayesian smoothing, image classification, machine learning, R.

1. Introduction

The use of remote sensing images is essential for environmental management. Satellite images have a wide range of applications, including measuring changes in land cover and assessing agriculture and natural habitats. Satellite measurements are the only viable mean to repeatedely survey vast regions, such as tropical forests and the polar regions. Additionally, remote sensing data plays a critical role in protecting the environment by supplying essential information to policymakers and conservationists.

Image classification is a key component of remote sensing image analysis. The goal of this task is to classify the pixels in an image into different categories based on their spectral characteristics, spatial patterns, or other pertinent features. Researchers employ machine learning methods like random forests (Belgiu and Dragut 2016) and deep learning (Ma *et al.* 2019) for image classification. The output of machine learning classifiers is a set of class probabilities for each pixel, organised in matrices that are identical in size to the original image. Each matrix provides the probability of a pixel's class membership. These matrices are the input to post-processing techniques that aim at refining the machine learning results. Post-processing leads to improved accuracy and better interpretability of the final output by reducing errors and minimizing noise (Schindler 2012).

Due to the complexity of remote sensing data, classification algorithms may inadvertently introduce noise or produce outliers. In medium-resolution satellite images, such as Landsat or Sentinel-2, pixels frequently display mixed spectral responses from different land cover types. Consequently, classification outcomes often contain misclassified pixels, particularly at the boundaries of two homogenous ground objects with distinct spectral characteristics. The aim of post-processing is to detect and rectify such inaccuracies, ensuring more accurate classification results.

The fragment of a Sentinel-2 image from in Rondonia, Brazil, shown in Figure 1, illustrates why post-processing is required for image classification. In the image, one can see the distinction between green forest areas and deforested areas depicted in shades of orange and brown. Um ponto importante pois é o main pitch para vender o paper na Introdução: Não sei se é daltonismo mas não vejo na Figura 1 pixels "in shades of orange and brown." è dito na introdução: " between green forest areas and deforested areas depicted in shades of orange and brown." Vejo apenas verde, rosa e purple.. The border pixels between green forest areas and brown bare soil have mixed responses. In many cases, such mixed pixels will be labelled as a third class which does not occur in this area. Thus, these border pixels are prone to misclassification, which needs to be corrected using suitable post-processing methods. Talvez circular com um desenho feito a mao as regioes problematicas possa ser util.

Image classification post-processing methods include Gaussian, semi-global, bilateral, and edge-aware filtering (Schindler 2012), modal filters (Ghimire *et al.* 2010), and co-occurrence matrices (Huang *et al.* 2014). All these methods have limitations to deal with boundary pixels. Both Gaussian and semi-global smoothing methods assume that each class has a uniform local variance value. These approaches assume probabilities change gradually within a window centred on each pixel, which is not the case for border pixels. As a result, these methods blur the crisp boundaries that separate homogenous regions in the image. Gaussian and semi-global smoothing techniques are ineffective in addressing the problem of mixed pixels at object boundaries (Huang *et al.* 2014). Bilateral smoothing uses two global parameters to maintain boundary shapes (Schindler 2012). The algorithm assumes that the spatial distribution of the class probabilities is isotropic in a pixel's

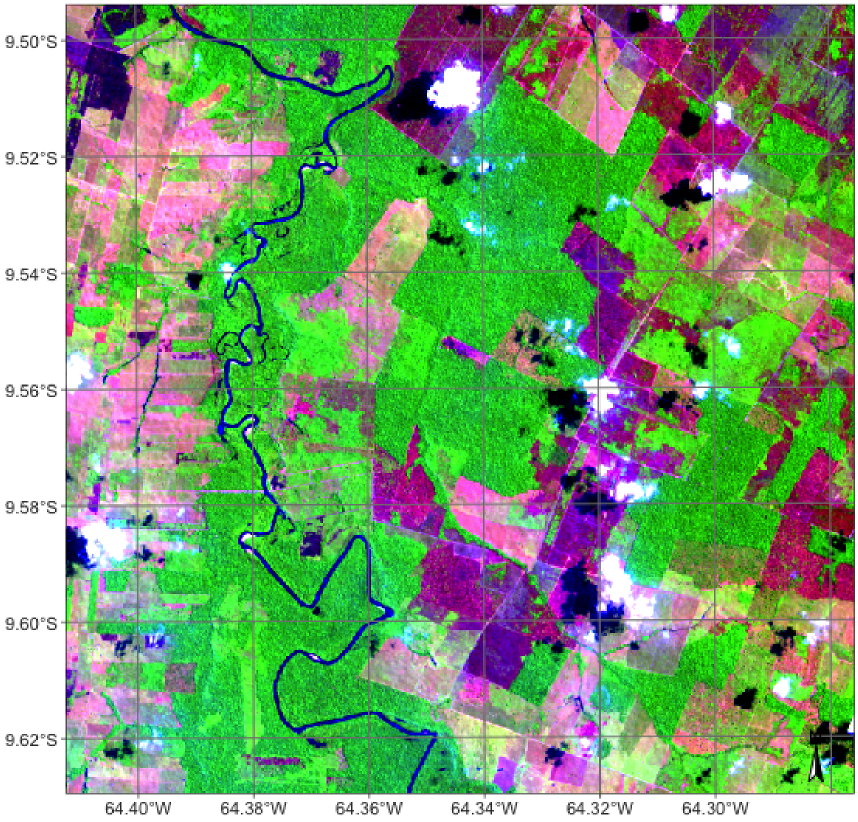

Figure 1: Detail of Sentinel-2 image in the Amazon forest.

neighbourhood. As we argue in this paper, such an assumption does not hold for border pixels. Hence, there is a need for post-processing methods that can handle situations involving mixed pixels and anisotropic neighborhoods.

The paper introduces a new post-processing algorithm for remote sensing image classification using an Empirical Bayes approach. The algorithm is available in the R package **bayesEO**. To simulate the impact of discontinuity between land classes, we use non-isotropic neighborhood definitions. Sugestao para a ultima sentenca: To represent empirically observed patterns in regions between land classes, we utilize non-isotropic neighborhood definitions. Our method allows the inclusion of expert knowledge to enhance the consistency of the resulting map. The **bayesEO** package can be combined with **sits**, an end-to-end toolkit for land use and land cover classification using big Earth observation data developed by the authors (Simoes *et al.* 2021).

Our package adds to other R libraries focused on the spatial data analysis, including **spdep** (Bivand and Pebesma 2023) and **rgeoda** (Li and Anselin 2022). **CARBayes** (Lee 2013) implements Bayesian models for spatial areal units. For image processing analysis, **terra** (Hijmans 2023) provides supervised classification with decision trees, while **stars** (Bivand and Pebesma 2023) includes functions for linear regression and random forest classification. For post-processing of machine learning image classification, **sits** also uses the method described in this paper. As far as we know, no current R package supports post-processing of image classification of remote sensing images. Therefore, the **bayesEO** is a useful addition to the facilities available in R for remote sensing image processing.

2. Methods

2.1. The land classification problem

In land classification, we deal with categorical data, with each category corresponding to a different land type (e.g., forest, grassland, wetland, crop). Land classification aims to subdivide the space into discrete areas, each associated with a distinct type of land use or cover. Borders between different land classes usually represent sharp transitions. Formally, the land classification problem can be expressed as follows. Given a set of n spatial locations or pixels $S = \{\mathbf{s}_1, \ldots, \mathbf{s}_n\}$, each with an associated d-dimensional feature vector $\mathbf{x}_1, \ldots, \mathbf{x}_s$ with values in $\mathscr{X} \subset \mathbb{R}^d$, we seek a classification function f mapping each pair $(\mathbf{s}_i, \mathbf{x}_i)$ to m land classes $K = \{1, ..., m\}$ probabilities:

$$
\begin{aligned}
f : (S, \mathscr{X}) &\longrightarrow K \\
(\mathbf{s}_i, \mathbf{x}_i) &\longmapsto (p_{i,1}, \ldots, p_{i,m})
\end{aligned}
\tag{1}
$$

The domain of f is not the cartesian product $(S \times \mathscr{X})$ because there is no point $(\mathbf{s}_i, \mathbf{x}_j)$ in the domain (when $i \neq j$).

with $0 \leq p_{i,k} \leq 1$ and $\sum_k p_{i,k} = 1 \; \forall i = 1, \ldots, n$. The value $p_{i,k}$ is interpreted as the probability that the i-th pixel belongs to the k-th class. The class assigned to the pixel is determined by the highest probability among the available options.

For pixel-based classification methods, the spatial location $\mathbf{s}_i$ of the pixel is not explicitly used by the classification function f. The reason is that f is obtained from a training dataset $T = \{\mathbf{s}_{t_1}, \ldots, \mathbf{s}_{t_s}\} \subset S$ with $s \ll n$. Each element $\mathbf{s}_{t_i} \in T$ is coupled with one single label $k_{t_i} \in K$ representing the class to which it belongs, even when the pixel has mixed classes on the ground. This training dataset is also called the *ground truth* dataset. Usually, it is composed of non-contiguous locations and therefore, there is no spatial neighborhood context available to train f. The only resource to fit f are the true labels and the features $\mathbf{x}$ measured at each location in the training dataset T.

The main idea behind our post-processing method is that a pixel-based classification should take into account its neighborhood pixels. Consider Figure 2 which shows a class assignment produced by a random forest algorithm on an image time series in the area shown in Figure 1. The image time series has been classified using the R package **sits**(Simoes *et al.* 2021).

The classified map has been produced by taking, for each pixel, the class of higher probability produced by the algorithm. The resulting map has many noisy areas with a high spatial variability of class assignments. This happens more frequently in two cases: (a) small clusters of pixels of one class inside a larger area of a different class; (b) transition zones between classes. In general, images of heterogeneous landscapes with high spatial variability have many mixed pixels, whose spectral response combines different types of land cover in a single ground resolved cell. For example, many pixels in the border between areas of classes `Forest` and `Clear_Cut_Bare_Soil` are wrongly assigned to the `Clear_Cut_Vegetation` class. This wrong assignment occurs because these pixels have a mixed response. Inside the ground cell captured by the sensor as a single pixel value, there are both trees and bare soil areas. Such results are undesirable and need to be corrected by post-processing.

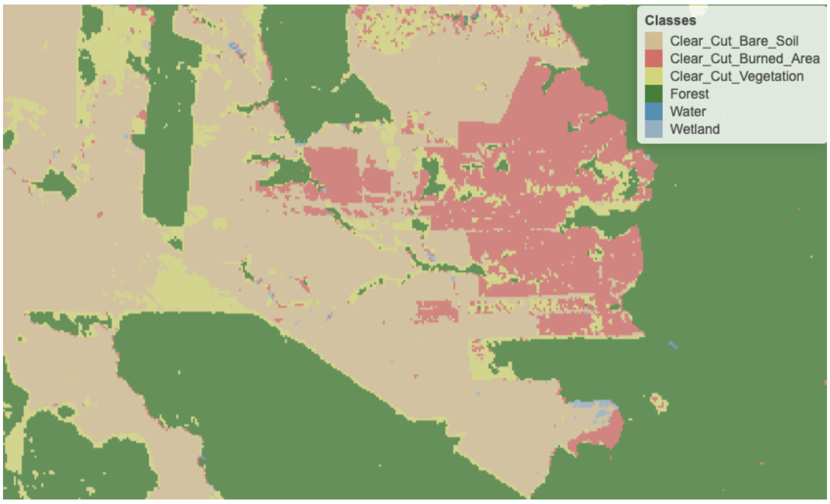

Figure 2: Detail of labelled map produced by pixel-based random forest without smoothing (source: authors)

To maintain consistency and coherence in our class representations, we should minimise small variations or misclassifications. We incorporate spatial coherence as a post-processing step to accomplish this. The probabilities associated with each pixel will change based on statistical inference, which depends on the values for each neighbourhood. Using the recalculated probabilities for each pixel, we get a better version of the final classified map.

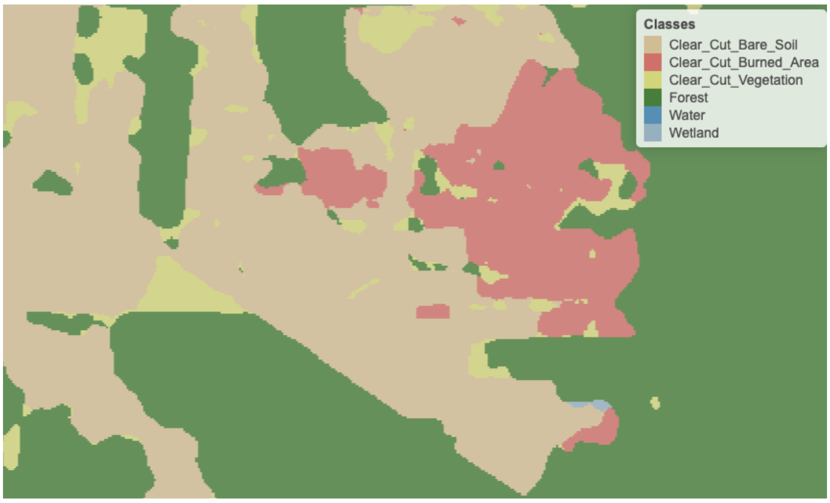

Figure 3: Detail of labelled map produced by pixel-based random forest after smoothing (source: authors)

Consider Figure 3, which is the result of Bayesian smoothing on the random forest algorithm outcomes. The noisy border pixels between two large areas of the same class have been removed. We have also removed small clusters of pixels belonging to one class inside larger areas of other classes. The outcome is a more uniform map, like the ones created through visual interpretation or object-based analysis. Details like narrow vegetation corridors or small forest roads might be missing in the smoothed image. However, the improved spatial consistency of the final map compensates for such losses, due to the removal of misclassified pixels that have mixed spectral responses.

2.2. A Bayesian approach to smooth image classification probabilities

We propose a Bayesian approach for post-processing of pixel probabilities. Let $\pi_{i,k} \geq 0$ be the prior probability of the i-th pixel belonging to class $k \in \{1, \dots, m\}$. Converting probabilities to the logit scale allows for less modelling restrictions. Accordingly, let

$$\mu_{i,k} = \log\left(\frac{\pi_{i,k}}{1-\pi_{i,k}}\right) \sim N(m_{i,k}, s^2_{i,k}) \tag{2}$$

Estava escrito $x_{i,k} = \log \dots$

The classification algorithm outputs the feature-based probabilities $(p_{i,1}, \dots, p_{i,m})$ from (1). We convert these observed values to the logit scale and assume a Gaussian distribution conditionally on $\mu_{i,k}$: $(x_{i,k}|\mu_{i,k}) = \log(p_{i,k}/(1-p_{i,k})) \sim N(\mu_{i,k}, \sigma^2_k)$. The variance σ^2_k will be estimated based on user expertise and taken as a hyperparameter to control the smoothness of the resulting estimate.

The standard Bayesian updating (Gelman *et al.* 2014) leads to the posterior distribution

$$(\mu_{i,k}|x_{i,k}) \sim \sum N\left(\frac{m_{i,k}\sigma^2_k + x_{i,k}s^2_{i,k}}{\sigma^2_k + s^2_{i,k}}, \left(\frac{1}{\sigma^2_k} + \frac{1}{s^2_{i,k}}\right)^{-1}\right) \tag{3}$$

Havia um $\sum$ antes da $N(\dots)$

The standard Bayesian method obtains the posterior distributions for each class probability, and the pixel is assigned to the class with higher posterior mean. The posterior mean is a weighted average between the pixel value $x_{i,k}$ and the prior mean $m_{i,k}$. When the prior variance $s^2_{i,k}$ is high, the algorithm assigns more weight to the pixel value $x_{i,k}$. Conversely, as the likelihood variance σ^2_k increases, the method assigns more weight to the prior mean $m_{i,k}$.

The fundamental idea behind Bayesian smoothing for land classification posits that image patches with similar characteristics usually have a dominant class. This dominant class has higher average probabilities and lower variance than other classes. A pixel assigned wrongly assigned?? misassigned?? to a different class will likely exhibit lower average probabilities and higher local variance in such regions. As a result, post-processing should adjust the class of this pixel to match the dominant class.

There is usually no prior information to specify $m_{i,k}$ and $s^2_{i,k}$. Because of that, we adopt an Empirical Bayes (EB) approach to obtain estimates of these prior parameters by considering the pixel neighborhood. However, using a standard symmetrical neighborhood for each pixel, based uniquely on the distance between locations, would not produce reasonable results for border pixels. For this reason, our EB estimates uses non-isotropic neighborhood, as explained below.

2.3. Empirical Bayes statistics using anisotropic neighbourhoods

Classification challenges arise for pixels located along the boundaries between areas containing different classes, as they exhibit characteristics of two or more classes simultaneously. In such cases, only a subset of the neighbors surrounding these boundary pixels belongs to the same class. To tackle this issue, we employ a non-isotropic definition of neighborhood to estimate the prior class distribution.

Pixels in the border between two areas of different classes pose a challenge. Only some of their neighbors belong to the same class as the pixel. Consider a boundary pixel with a neighborhood defined by a 7 x 7 window, located along the border between the `Forest` and `Grassland` classes. To estimate the prior probability of the pixel being a `Forest`, we should only take into account the neighbours on one side of the border that are likely to be correctly classified as `Forest`. Pixels on the opposite side of the border should be disregarded, since they are unlikely to belong to the same spatial process. In practice, we use only half of the pixels in the 7 x 7 window, opting for those that have a higher probability of being `Forest`. For the prior probability of the `Grassland` class, we reverse the selection and only consider those on the opposite side of the border.

Although this choice of neighbourhood may seem unconventional, it is consistent with the assumption of non-continuity of the spatial processes describing each class. A dense forest patch, for example, will have pixels with strong spatial autocorrelation for values of the `Forest` class; however, this spatial autocorrelation doesn't extend across its border with other land classes.

Thus, the EB estimates uses specific neighbourhoods $\mathscr{N}_{i,k}$ for each class k and pixel i. We use an L-statistic to estimate $m_{i,k}$ and $s^2_{i,k}$ in our EB approach. Let $\alpha \in (0,1)$ and W_i be the set of w nearest neighbors of pixel i (excluding the i-th pixel itself). Also, let $\mathbb{F}_{i,k}$ be the empirical distribution of the $x_{j,k}$ for $j \in W_i$. Then, we take

$$\hat{m}_{i,k} = \frac{1}{1-\alpha} \int_{\alpha}^{\infty} \mathbb{F}^{-1}_{i,k}(s)\, ds\,, \tag{4}$$

the average of the largest (1—α)-th fraction order statistics of the $p_{i,k}$ logit transformed observations. Likewise, based on the these same $(1-\alpha)$-th subset, we obtain an empirical estimate $\hat{s}^2_{i,k}$. The values of $\hat{m}_{i,k}$ and $\hat{s}^2_{i,k}$ are used in the Bayesian updating (cf Equation 3).

2.4. Effect of the hyperparameter

The parameter σ^2_k controls the level of smoothness. If σ^2_k is zero, the estimated value $E[\mu_{i,k}|x_{i,k}]$ will be the pixel value $x_{i,k}$. Values of the likelihood variance σ^2_k, which are small relative to the prior variance $s^2_{i,k}$ increase our confidence in the original probabilities. Conversely, likelihood variances σ^2_k, which are large relative to the prior variance $s^2_{i,k}$, increase our confidence in the average probability of the neighborhood.

Thus, the parameter σ^2_k expresses confidence in the inherent variability of the distribution of values of a class k. The smaller the parameter σ^2_k, the more we trust the estimated probability values produced by the classifier for class k. Conversely, higher values of σ^2_k indicate lower confidence in the classifier outputs and improved confidence in the local average values.

Consider the following two-class example. Take a pixel i with probability 0.4 (logit $x_{i,1} = —0.4054$) for class A, and probability 0.6 (logit $x_{i,2} = 0.4054$) for class B. Without post-processing, the pixel will be labelled as class B. Consider a local average of 0.6 (logit $m_{i,1} = 0.4054$) for class A and 0.4 (logit $m_{i,2} = —0.4054$) for class B. This is an outlier classified as class B in the midst of a set of pixels of class A.

Given this situation, we apply the proposed method. Suppose the local variance of logits to be $s_{i,1}^2 = 5$ for class A and $s_{i,2}^2 = 10$ and for class B. This difference is to be expected if the local variability of class A is smaller than that of class B. To complete the estimate, we need to set the parameter σ_k^2, representing our belief in the variability of the probability values for each class.

Setting σ_k^2 will be based on our confidence in the local variability of each class around pixel i. If we considered the local variability to be high, we can take both σ_1^2 for class A and σ_2^2 for class B to be both 10. In this case, the Bayesian estimated probability for class A is 0.52 and for class B is 0.48 and the pixel will be relabelled as being class A.

By contrast, if we consider local variability to be high If we set σ^2 to be 5 for both classes A and B, the Bayesian probability estimate will be 0.48 for class A and 0.52 for class B. In this case, the original class will be kept. Therefore, the result is sensitive to the subjective choice of the hyperparameter.

3. Software and examples

3.1. The BayesEO package

The post-processing method described in this paper is implemented in the R **bayesEO**, which is available on CRAN and also on github. The examples below also require installation of package **sitsdata**. Because of its size, they cannot be loaded into CRAN; it has to be obtained directly for its github repository, The code below retrieves and loads **bayesEO** and **sitsdata**.

```
R> # configuration
R> # if bayesEO is not installed, install it
R> if (!requireNamespace("bayesEO", quietly = TRUE))
+     devtools::install_github("e-sensing/bayesEO")
R> library(bayesEO)
R> # if sitsdata is not installed, get it
R> if (!requireNamespace("sitsdata", quietly = TRUE))
+     devtools::install_github("e-sensing/sitsdata")
R> library(sitsdata)
```

3.2. Reading a probability data cube

The input for post-classification is an image with probabilities produced by a machine learning algorithm. This image should be a single file with multiple bands, where each band contains the pixel probabilities of a single class. In the examples, we use a file produced by a random forest algorithm applied to a data cube of Sentinel-2 images for tile “20LLQ” in the period 2020-06-04 to 2021-08-26. The file uses an INT2S data type with integer values between [0..10000] to represent probabilities ranging from 0 ro 1.

The training data has six classes: (a) `Forest` for natural tropical forest; (b) `Water` for lakes and rivers; (c) `Wetlands` for areas where water covers the soil in the wet season; (d) `ClearCut_Burn` for areas where fires cleared the land after tree removal; (e) `ClearCut_Soil` where the forest has been completely removed; (f) `ClearCut_Veg` where some vegetation remains after most trees have been removed.

To read the probability image, we use the function `bayes_read_probs()` whose parameters are an image file with probabilities and a set of labels. The probability image can be viewed using `bayes_plot_probs()`. The function `bayes_label()` takes the probability image as an input and produces a categorical map which associates each pixel with the class of highest probability.The map is rendered using `bayes_plot_map()`.

```
R> # select the file containing the class probability
R> data_dir <- system.file("/extdata/Rondonia-20LLQ/", package = "sitsdata")
R> probs_file <- paste0(data_dir,
+                        "SENTINEL-2_MSI_20LLQ_2020-06-04_2021-08-26_probs_v1.tif")
R> # set the labels
R> labels <- c("Water", "ClearCut_Burn", "ClearCut_Soil",
+              "ClearCut_Veg", "Forest", "Wetland")
R> # read the probability image
R> probs_image <- bayes_read_probs(probs_file, labels)
R> # plot the probabilities for classes Forest and ClearCut_Soil
R> bayes_plot_probs(probs_image, labels = c("Forest", "ClearCut_Soil",
+                                            "ClearCut_Veg", "ClearCut_Burn"))
```

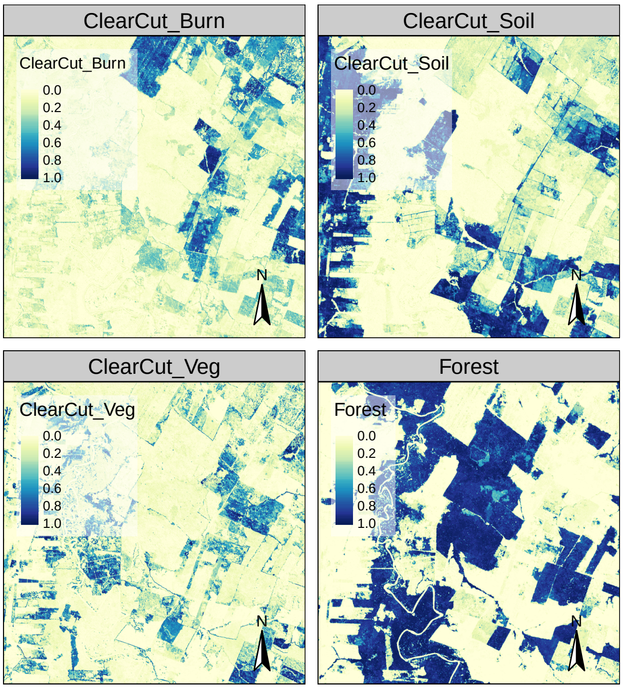

Figure 4: Class probabilities produced by random forest algorithm

Figure 4 shows the plot of the probability images for classes `Forest`, `ClearCut_Soil`, `ClearCut_Veg` and `ClearCut_Burn`. The map for class `Forest` shows high probability values associated with compact patches. Even inside relatively homogenous areas, there are locations whose probabilities vary significantly. There is large variation in areas close to the river that runs from South to

North, corresponding to riparian forest of lower density. Class ClearCut_Soil is mostly composed of dense areas of high probability whose geometrical boundaries result from forest cuts. Some of the patches of this class, as those in the upper-left corner, have high variation in their probabilities. The class ClearCut_Veg is less well-defined than the others; this is to be expected since this is a transitional class between a natural forest and areas of bare soil. Patches associated to class ClearCut_Brun include both homogeneous areas of high probability and areas of mixed response. Overall, there are many areas with mixed responses, which highlight the value of spatial smoothing.

```
R> # produce a labelled map
R> map_no_smooth <- bayes_label(probs_image)
R> # show the map
R> bayes_plot_map(map_no_smooth)
```

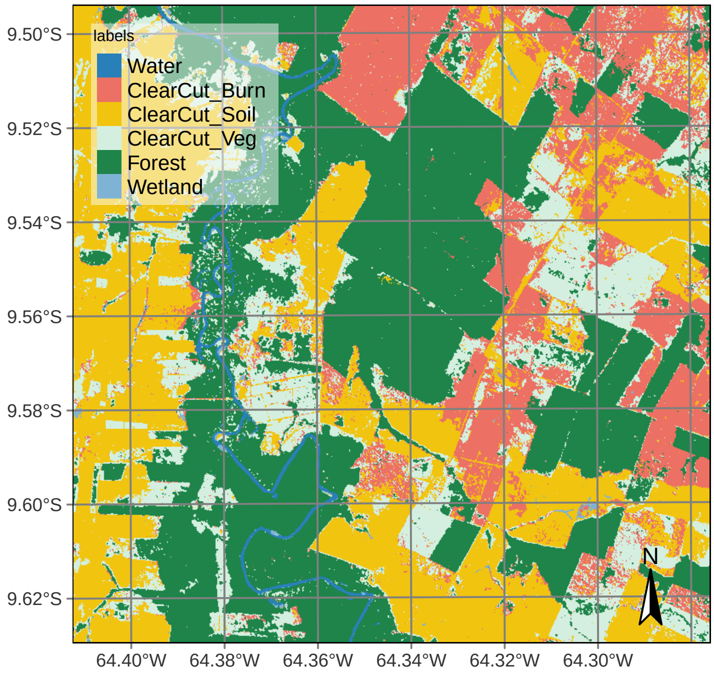

Figure 5: Labelled map without smoothing.

Figure 5 shows the classification map, obtained by taking the class of higher probability to each pixel, without considering the spatial context. The non-smoothed labelled map shows the need for post-processing, since it contains a significant number of outliers and areas with mixed labelling.

3.3. Estimating the local logit variances

The local logit variances correspond to the $s^2_{i,k}$ parameter in the Bayesian inference and are estimated by bayes_variance(). Its main parameters are: (a) a SpatRaster object; (b) window_size, dimension of the local neighbourhood; (c) neigh_fraction, the percentage of pixels in the neighbourhood used to calculate the variance. The example below uses half of the pixels of a 7 × 7 window to estimate the variance. The chosen pixels will be those with the highest probability pixels to be more representative of the actual class distribution. The output values are the logit variances in the vicinity of each pixel.

```
R> var_image <- bayes_variance(
+      x = probs_image,
+      window_size = 7,
+      neigh_fraction = 0.50)
R> bayes_summary(var_image)
```

```
 Water              ClearCut_Burn      ClearCut_Soil      ClearCut_Veg
 Min.   : 0.0000    Min.   :0.001275   Min.   :0.001253   Min.   : 0.0000
 1st Qu.: 0.1328    1st Qu.:0.056433   1st Qu.:0.080421   1st Qu.: 0.0953
 Median : 0.5077    Median :0.118231   Median :0.181310   Median : 0.2201
 Mean   : 2.0516    Mean   :0.236608   Mean   :0.450867   Mean   : 0.5048
 3rd Qu.: 4.2253    3rd Qu.:0.255588   3rd Qu.:0.396300   3rd Qu.: 0.5327
 Max.   :31.2500    Max.   :9.864704   Max.   :9.295169   Max.   :14.7009
 Forest              Wetland
 Min.   : 0.00000   Min.   : 0.0000
 1st Qu.: 0.06845   1st Qu.: 0.0553
 Median : 0.21232   Median : 0.1273
 Mean   : 1.09353   Mean   : 0.3752
 3rd Qu.: 1.15513   3rd Qu.: 0.2922
 Max.   :18.80040   Max.   :10.4210
```

The choice of the 7 × 7 window size is a compromise between having enough values to estimate the parameters of a normal distribution and the need to capture local effects for class patches of small sizes. Classes such as `Water` and `ClearCut_Burn` tend to be spatially limited; a bigger window size could result in invalid values for their respective normal distributions.

The summary statistics show that most local variance values are low, which is an expected result. Areas of low variance correspond to pixel neighborhoods of high logit values for one of the classes and low logit values for the others. High values of the local variances are relevant in areas of confusion between classes. Figure 6 shows the values of local logit variances for classes `ClearCut_Soil` and `Forest`, considering only the 4^{th} quartile of the distribution. Only the top 25% of the values for each class are shown, emphasizing areas of high local variability.

```
R> bayes_plot_var(var_image,
+                   quantile = 0.75,
+                   palette = "Greens",
+                   labels = c("ClearCut_Veg", "Forest"))
```

Comparing the logit variance maps of Figure 6 with the probability maps of Figure 4 emphasizes the relevance of expert knowledge. The areas of high probability of class `Forest` are mostly made of compact patches. However, there are areas of high local variance inside these patches, due to natural variation of the forest and also resulting from forest degradation. By contrast, class `ClearCut_Veg` represents a transition between natural forest areas and places where all trees have

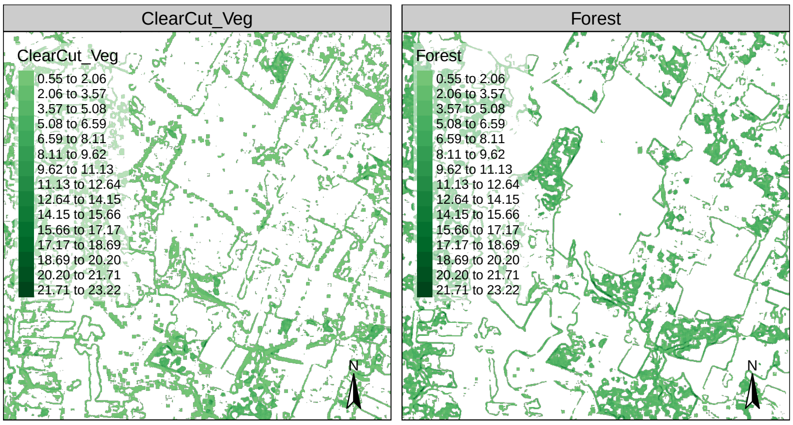

Figure 6: Logit variance map showing values above the 3rd quartile for classes ClearCutVeg and Forest.

been cut, which are associated to class `ClearCut_Soil`. Class `ClearCut_Veg` has a high spectral variability, since the extent of remaining vegetation after most trees have been removed is not uniform. For this reason, many areas of high local variance of class `ClearCut_Veg` are located in the borders between homogenous areas.

3.4. Applying Bayesian smoothing to remove outliers

As discussed above, the effect of the Bayesian estimator depends on the values of the a priori variance σ_k^2 set by the user and the neighbourhood definition to compute the local variance $s_{i,1}^2$ for each pixel. We set σ^2 values for each class by considering values in the 4th quartile, close to the maximum local logit variance for each class. To remove the outliers in the classification map, **bayesEO** provides `bayes_smooth()`. Its main parameters are: (a) `x`, a probability image; (b) `window_size`, dimension of the local neighbourhood; (c) `smoothness`, prior logit variances for each class.

```
R> smooth_probs <- bayes_smooth(
+     probs_image,
+     window_size = 7,
+     smoothness = c(
+       "Water" = 15, "ClearCut_Burn" = 5, "ClearCut_Soil" = 6,
+       "ClearCut_Veg" =  8, "Forest" = 10, "Wetland" = 5)
+ )
```

The impact of Bayesian smoothing can be best captured by producing a labelled map using `bayes_label()`, taking the smoothed image as its input. Figure 7 shows that the outliers and isolated pixels have been removed.

```
R> map_smooth_bayes <- bayes_label(smooth_probs)
R> bayes_plot_map(map_smooth_bayes)
```

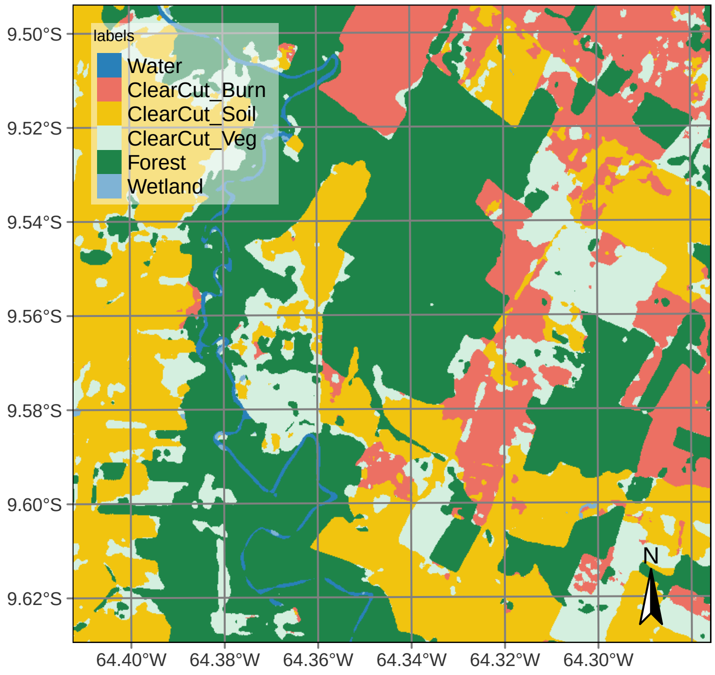

Figure 7: Labeled map with Bayesian smoothing.

In the smoothed map, the outliers have been removed by expanding `Forest` areas. Forests have replaced small corridors of water and soil encircled by trees. This effect is due to the high probability of forest detection in the training data. To better assert the impact of smoothing, consider the class areas in the non-smoothed and smoothed maps.

```
R> # get the summary of the non-smoothed map
R> sum1 <- bayes_summary(map_no_smooth)
R> colnames(sum1) <- c("class", "area_k2_no_smooth")
R> # get the summary of the smoothed map
R> sum2 <- bayes_summary(map_smooth_bayes)
R> colnames(sum2) <- c("class", "area_k2_smooth")
R> # compare class areas of non-smoothed and smoothed maps
R> dplyr::inner_join(sum1, sum2, by = "class")
```

```
# A tibble: 6 x 3
  class         area_k2_no_smooth area_k2_smooth
  <chr>                     <dbl>          <dbl>
1 Water                      2.55           2.31
2 ClearCut_Burn             34.6           32.8
3 ClearCut_Soil             65.7           68.3
4 ClearCut_Veg              37.3           33.2
5 Forest                    83.8           88.0
6 Wetland                      0.99           0.31
```

3.5. Relevance of expert knowledge in Bayesian inference

In the smoothed map (Figure 7), the most frequent classes (`ClearCut_Soil` and `Forest`) increased their areas at the expense of the others. As shown in Figure 4, these classes occur in more compact patches than the others. Based on the experience of the authors with different experts on land use classification, there are two main approaches for setting the σ_k^2 parameter:

1. Increase the neighborhood influence compared with the probability values for each pixel, setting high values (20 or above) to σ_k^2 and increasing the neighborhood window size. Classes whose probabilities have strong spatial autocorrelation will tend to replace outliers.
2. Reduce the neighborhood influence compared with the probabilities for each pixel of class k, setting low values (five or less) to σ_k^2. In this way, classes with low spatial autocorrelation are more likely to keep their original labels.

Consider the case of forest areas and watersheds. If an expert wishes to have compact areas classified as forests without many outliers inside them, she will set the σ^2 parameter for the class `Forest` to be high. For comparison, to avoid that small watersheds with few similar neighbors being relabeled, it is advisable to avoid a strong influence of the neighbors, setting σ^2 to be as low as possible. Therefore, the choice of σ^2 depends on the effect intended by the expert in the final classified map.

4. Comparison with other methods

To better understand the benefits of Bayesian smoothing, in this section we present the results of post-processing using two alternatives: Gaussian smoothing and bilateral smoothing. Gaussian smoothing is a low-pass filter where the weights are based on the normal distribution. Pixels near the center of the kernel have a higher weight, and the weight decreases for pixels further away from the center. This creates a blurring effect that is stronger at the center and weaker at the edges. The extent of smoothing is controlled by the standard deviation of a Gaussian distribution. The convolution operation effectively averages pixel values in the neighborhood defined by the kernel, with weights given by the Gaussian distribution. This results in a reduction of high-frequency noise and a blurring effect.

The Gaussian Kernel is expressed as

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{5}$$

where σ is the standard deviation of the Gaussian distribution. The convolution for Gaussian smoothing is expressed as

$$I'(x,y) = \sum_{i=-k}^{k} \sum_{j=-k}^{k} G(i,j) \cdot I(x-i, y-j) \tag{6}$$

In this equation, $I'(x,y)$ represents the smoothed image, $I(x,y)$ is the original image, $G(i,j)$ is the Gaussian kernel, and (i,j) are the indices running over the kernel size, typically centered around zero. The following code computes a Gaussian smoothing filter, labels the image and plots the resulting map.

```
R> # compute Gaussian smoothing, label and plot map
R> gaussian_smooth(probs_image, window_size = 7, sigma = 5 ) |>
+    bayes_label() |>
+    bayes_plot_map()
```

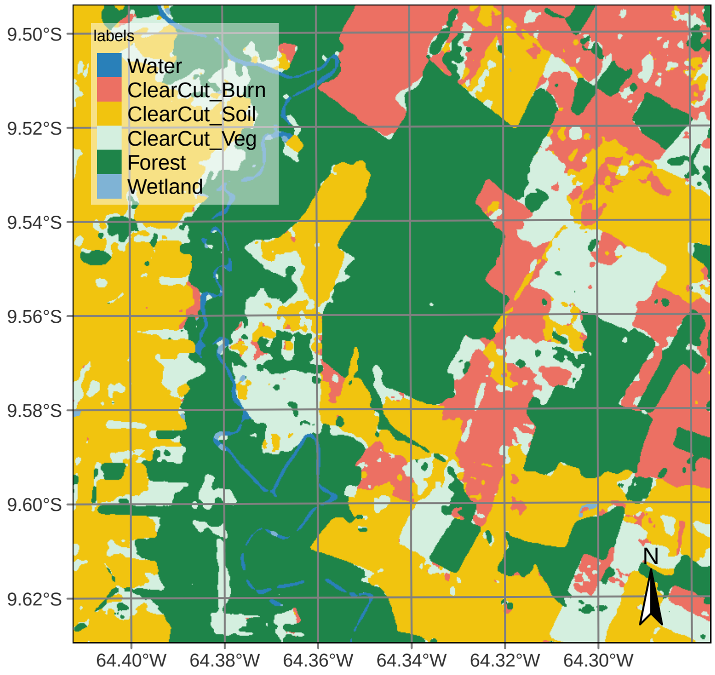

Figure 8: Labeled map using Gaussian smoothing.

Comparison between Figure 7 and Figure 8 shows subtle but important differences. Bayesian smoothing does a better job of preserving edges and small homogeneous areas than Gaussian filtering. The result is a more consistent map. Parameters for Gaussian smoothing are the same for each class and assume isotropic neighbourhoods, while Bayesian inference allows the expert to control the output. Moreover, the parameter σ for Gaussian filtering was set by trial-and-error. The above example shows the best result after trying different alternatives.

Bilateral smoothing aims to reduce noise while preserving edges. Unlike the Gaussian filter, which can blur edges, the bilateral filter maintains sharp edges by taking into account the difference in intensity values between pixels (Tomasi and Manduchi 1998). Bilateral filtering combines a spatial filter, like the Gaussian filter, which considers the spatial closeness of pixels, with a range filter, which considers the similarity in intensity values between pixels. By considering intensity differences, the bilateral filter can preserve edges.

The bilateral filter has two main parameters: (a) the spatial parameter σ controls the spatial extent of the kernel; (b) the range parameter τ controls how much a pixel must differ in intensity for it to be considered different. Its mathematical expression is

$$I'(x,y) = \frac{1}{W(x,y)} \sum_{i=-k}^{k} \sum_{j=-k}^{k} I(x-i, y-j) \cdot G_{\sigma}(i,j) \cdot G_{\tau}(I(x-i,y-j) - I(x,y)) \tag{7}$$

where $I'(x,y)$ is the filtered image and $I(x,y)$ is the original image. $G_\sigma(i,j)$ is the spatial Gaussian function, which decreases with distance from the central pixel, controlled by the spatial standard deviation σ. $G_\tau(I(x+i,y+j)-I(x,y))$ is the range Gaussian function, which decreases with the intensity difference between the neighboring pixel and the central pixel controlled by the range standard deviation τ.

$W(x,y)$ is a normalization factor defined as

$$W(x,y) = \sum_{i=-k}^{k} \sum_{j=-k}^{k} G_{\sigma_s}(i,j) \cdot G_{\sigma_r}(I(x-i,y-j)-I(x,y)) \tag{8}$$

This formula illustrates how the bilateral filter considers both spatial proximity and intensity similarity in its smoothing process, thereby preserving edges while reducing noise in other areas. The following code performs bilateral smoothing.

```
R> # compute Gaussian smoothing
R> bilat_smooth <- bilateral_smooth(
+     probs_image,
+     window_size = 7,
+     sigma = 5,
+     tau = 2.0
+ )
R> # produce the labelled map
R> bilat_map <- bayes_label(bilat_smooth)
R> # plot the map produced by Bilateral smoothing
R> bayes_plot_map(bilat_map)
```

The resulting map after bilateral smoothing is similar to the output of Bayesian smoothing. However, this result required some rounds of trial and error to set the best values for the σ and τ parameters. There is no objective method for choosing the appropriate values for them. By contrast, the smoothness parameters for Bayesian filtering are select based on the variances of the class probabilities.

This example shows the value of the Bayesian inference procedure compared with smoothing methods such as Gaussian and edge-aware filtering (Schindler 2012). Most post-classification procedures use ad-hoc parameters which are not directly linked to the properties of the data. These parameters are based on the structure of the algorithm (e.g, size of the Gaussian kernel), not being easily defined separately for each class. Bayesian inference allows the expert to control the output and to set the parameters based on objective criteria.

5. Using smoothing in connection with time-first, space-later methods

Time-first, space-later is a concept in satellite image classification that takes time series analysis as the first step for analysing remote sensing data, with spatial information being considered after all time series are classified (Camara *et al.* 2016). Satellite image time series are calibrated and comparable measures of the same location on Earth at different times. When associated with frequent revisits, image time series can capture significant land use and land cover changes.

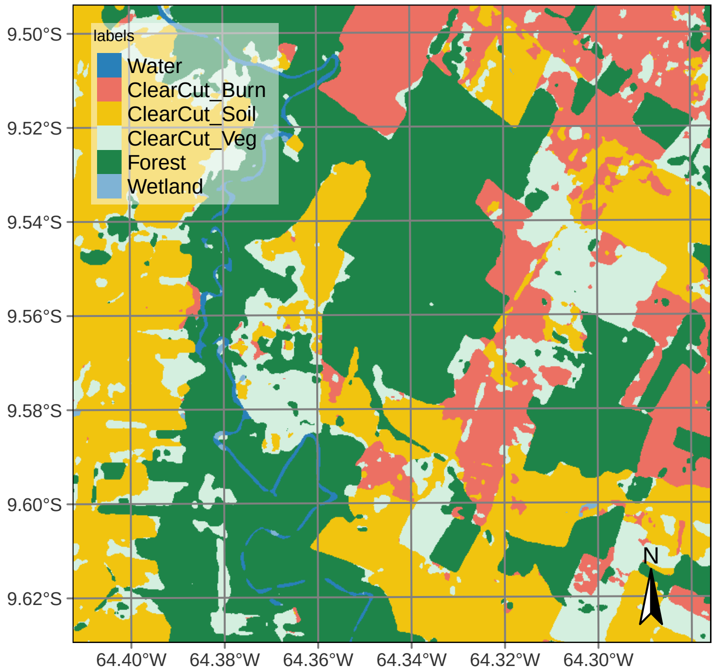

Figure 9: Labeled map using bilateral smoothing.

Detecting and tracking seasonal and long-term trends becomes feasible, as well as identifying anomalous events or patterns in the data, such as wildfires, floods, or droughts Woodcock *et al.* (2020).

This approach takes image time series as the first step for analysing remote sensing data. The *time-first* part brings a better understanding of changes in landscapes. Detecting and tracking seasonal and long-term trends becomes feasible, as well as identifying anomalous events or patterns in the data, such as wildfires, floods, or droughts. Each pixel in a data cube is treated as a time series, using information available in the temporal instances of the case. Time series classification is pixel-based, producing a set of labeled pixels. This result is then used as input to the *space-later* part of the method. In this phase, Bayesian algorithm improves the results of time-first classification by considering the spatial neighbourhood of each pixel. The resulting map thus combines both spatial and temporal information.

In R, the **sits** package combines machine learning methods for time series classification with spatial-based smoothing(Simoes *et al.* 2021). It has been used for large-scale land use classification in Brazil Picoli *et al.* (2018). A recent comparative study shows that image time series classification with spatial smoothing achieves better results than purely spatial-based methods(Hadi *et al.* 2023). Thus, Bayesian smoothing is a key component of the time-first, space-later Earth observation data analysis.

6. Conclusion

This paper presents a new method for post-processing of Earth observation images that have been classified using machine learning algorithms. Unlike ad-hoc smoothing methods avaliable in the literature, the proposed algorithm relies on objective metrics of class probability variance. We propose an Empirical Bayes approach to post-processing, which has a sound mathematical basis. The method has been validated for large-scale data analysis in connection with a time-first, space-based approach and represents a worthy alternative to other post-processing methods available in the literature.

Acknowledgments

The authors acknowledge the support of the following institutions: (a) Amazon Fund, established by Brazil with financial contribution from Norway, through contract 17.2.0536.1.; (b) International Climate Initiative of the Germany Federal Ministry for the Environment, Nature Conservation, Building and Nuclear Safety (IKI) under grant 17-III-084-Global-A-RESTORE+ ("RESTORE+: Addressing Landscape Restoration on Degraded Land in Indonesia and Brazil"); (c) Microsoft Planetary Computer initiative under the GEO-Microsoft Cloud Computer Grants Programme; (d) Instituto Clima e Sociedade, under the project grant "Modernization of PRODES and DETER Amazon monitoring systems" (grants G-22-01193 and G-23-01643); (e) Open-Earth-Monitor Cyberinfrastructure project, under European Commission grant agreement No. 101059548.

References

Belgiu M, Dragut L (2016). "Random Forest in Remote Sensing: A Review of Applications and Future Directions." *ISPRS Journal of Photogrammetry and Remote Sensing*, **114**, 24–31.

Bivand R, Pebesma E (2023). *Spatial Data Science: With Applications in R*. Chapman and Hall/CRC, New York. ISBN 978-0-429-45901-6. doi:10.1201/9780429459016.

Camara G, Assis LF, Ribeiro G, Ferreira KR, Llapa E, Vinhas L (2016). "Big Earth Observation Data Analytics: Matching Requirements to System Architectures." In *5th ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data*, pp. 1–6. ACM, Burlingname, CA, USA.

Gelman A, Carlin JB, Stern HS, Dunson DB, Vehtari A, Rubin DB (2014). *Bayesian Data Analysis, Third Edition*. CRC Press. ISBN 978-1-4398-4095-5.

Ghimire B, Rogan J, Miller J (2010). "Contextual Land-Cover Classification: Incorporating Spatial Dependence in Land-Cover Classification Models Using Random Forests and the Getis Statistic." *Remote Sensing Letters*, **1**(1), 45–54.

Hadi F, Sabri LM, Prasetyo Y, Sudarsono B (2023). "Leveraging Time-Series Imageries and Open Source Tools for Enhanced Land Cover Classification." In *IOP Conference Series: Earth and Environmental Science*, volume 1276, p. 012035. IOP Publishing. doi:10.1088/1755-1315/1276/1/012035.

Hijmans R (2023). "Spatial Data Science with R and "Terra" — R Spatial." *Technical report*.

Huang X, Lu Q, Zhang L, Plaza A (2014). "New Postprocessing Methods for Remote Sensing Image Classification: A Systematic Study." *IEEE Transactions on Geoscience and Remote Sensing*, **52**(11), 7140–7159.

Lee D (2013). "CARBayes: An R Package for Bayesian Spatial Modeling with Conditional Autoregressive Priors." *Journal of Statistical Software*, **55**, 1–24. ISSN 1548-7660. doi:10.18637/jss.v055.i13.

Li X, Anselin L (2022). *Rgeoda: R Library for Spatial Data Analysis*.

Ma L, Liu Y, Zhang X, Ye Y, Yin G, Johnson BA (2019). "Deep Learning in Remote Sensing Applications: A Meta-Analysis and Review." *ISPRS Journal of Photogrammetry and Remote Sensing*, **152**, 166–177. ISSN 0924-2716. doi:10.1016/j.isprsjprs.2019.04.015.

Picoli M, Camara G, Sanches I, Simoes R, Carvalho A, Maciel A, Coutinho A, Esquerdo J, Antunes J, Begotti RA, Arvor D, Almeida C (2018). "Big Earth Observation Time Series Analysis for Monitoring Brazilian Agriculture." *ISPRS journal of photogrammetry and remote sensing*, **145**, 328–339. doi:10.1016/j.isprsjprs.2018.08.007.

Schindler K (2012). "An Overview and Comparison of Smooth Labeling Methods for Land-Cover Classification." *IEEE transactions on geoscience and remote sensing*, **50**(11), 4534–4545.

Simoes R, Camara G, Queiroz G, Souza F, Andrade PR, Santos L, Carvalho A, Ferreira K (2021). "Satellite Image Time Series Analysis for Big Earth Observation Data." *Remote Sensing*, **13**(13), 2428. doi:10.3390/rs13132428.

Tomasi C, Manduchi R (1998). "Bilateral Filtering for Gray and Color Images." In *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, pp. 839–846. Narosa Publishing House, Bombay, India. ISBN 978-81-7319-221-0. doi:10.1109/ICCV.1998.710815.

Woodcock CE, Loveland TR, Herold M, Bauer ME (2020). "Transitioning from Change Detection to Monitoring with Remote Sensing: A Paradigm Shift." *Remote Sensing of Environment*, **238**, 111558. ISSN 00344257. doi:10.1016/j.rse.2019.111558.

Affiliation:

Gilberto Camara
Nat Inst for Space Research
Brazil
Avenida dos Astronautas, 1758
12227-001 Sao Jose dos Campos, Brazil
E-mail: gilberto.camara"'inpe.br

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:10.18637/jss.v000.i00 *Accepted:* yyyy-mm-dd