



Bayesian inference for post-processing of remote sensing image classification

Gilberto Camara 

Nat Inst for Space Research
Brazil

Renato Assunção 

Federal Univ of Minas Gerais
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

A key component of remote sensing image analysis is image classification, which aims to categorize the pixels in an image into different classes using machine learning methods. The result of machine learning classifiers is a set of class probabilities for each pixel. These class probabilities are the input to post-processing techniques that aim to improve on the machine learning results. This paper introduces a new post-processing algorithm that uses an Empirical Bayes approach. To simulate the impact of discontinuity between land classes, we use non-isotropic neighborhood definitions. Our method allows the inclusion of expert knowledge to enhance the consistency of the resulting map. The method has been validated for large-scale data analysis in connection with a time-first, space-based approach and is available in the R package **bayesEO**.

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 repeatedly survey vast regions, such tropical forests and the polar regions. Remote sensing data plays a critical role in protecting the environment by supplying essential information to policymakers and conservationists.

Image classification is an key component of remote sensing image analysis. The goal of this task is to categorize the pixels in an image into different classes based on their spectral characteristics, spatial patterns, or other relevant features. Researchers employ machine learning methods like random forests (Belgiu and Dragut 2016) and deep learning (Ma *et al.* 2019) in image classification. The result 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 to improve on 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 can introduce noise or produce outliers. Spectral responses of ground targets have large variability. Pixels in medium-resolution satellite images, like Landsat or Sentinel-2, often have mixed spectral responses from different land cover types. For these reasons, classifiers produce results that include misclassified pixels, especially in the boundaries between two homogenous ground objects which have different spectral responses. Post-processing methods need to be able to detect and correct such errors.

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. 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.

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 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

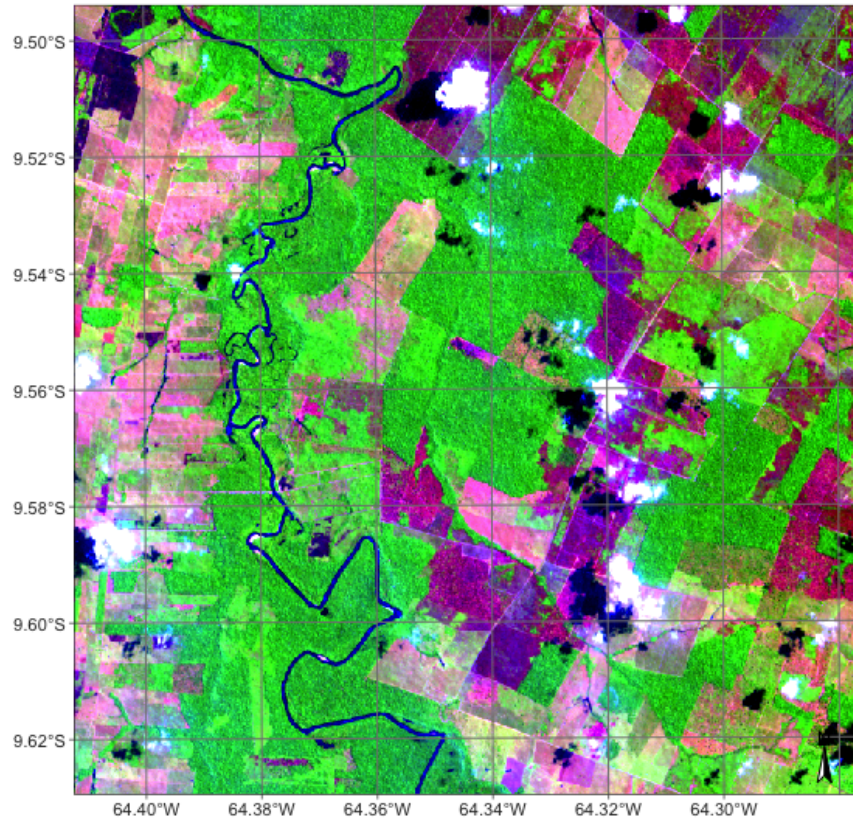


Figure 1: Detail of Sentinel-2 image in the Amazon forest.

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 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 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. For this reason, we consider **bayesEO** to be 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 $S = \{\mathbf{s}_1, \dots, \mathbf{s}_n\}$, each with an associated d -dimensional feature vector $\mathbf{x}_1, \dots, \mathbf{x}_s$ with values in $\mathcal{X} \subset \mathbb{R}^m$ and a set of m land classes $K = \{1, \dots, m\}$, we seek a classification function f such that

$$\begin{aligned} f: S \times \mathcal{X} &\longrightarrow K \\ (\mathbf{s}_i, \mathbf{x}_i) &\longmapsto (p_{i,1}, \dots, p_{i,m}) \end{aligned} \quad (1)$$

with $0 \leq p_{i,k} \leq 1$ and $\sum_k p_{i,k} = 1 \forall i = 1, \dots, n$. 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 \mathbf{s}_i dimension is not explicitly used by the classification function f . The reason is that f is obtained from a training dataset $T = \{\mathbf{s}_{t_1}, \dots, \mathbf{s}_{t_s}\} \subset S$ 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 features \mathbf{x} measured at each location in the training dataset T .

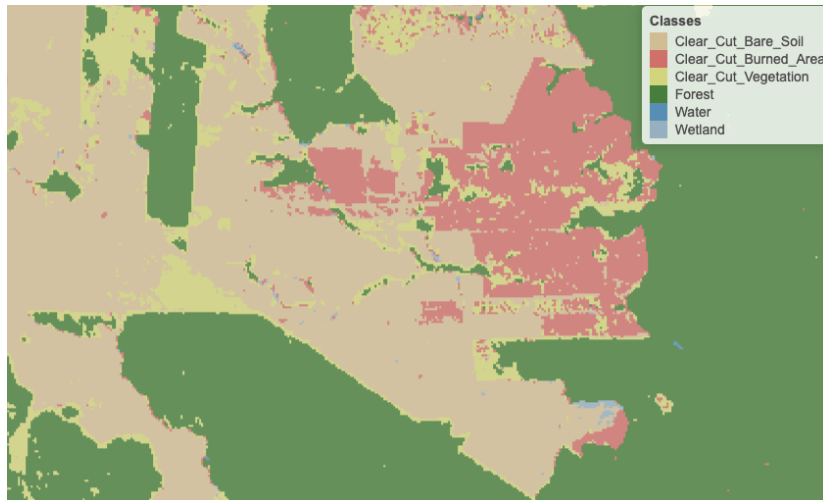


Figure 2: Detail of labelled map produced by pixel-based random forest without smoothing (source: authors)

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 a image time series in a subset of 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.

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.

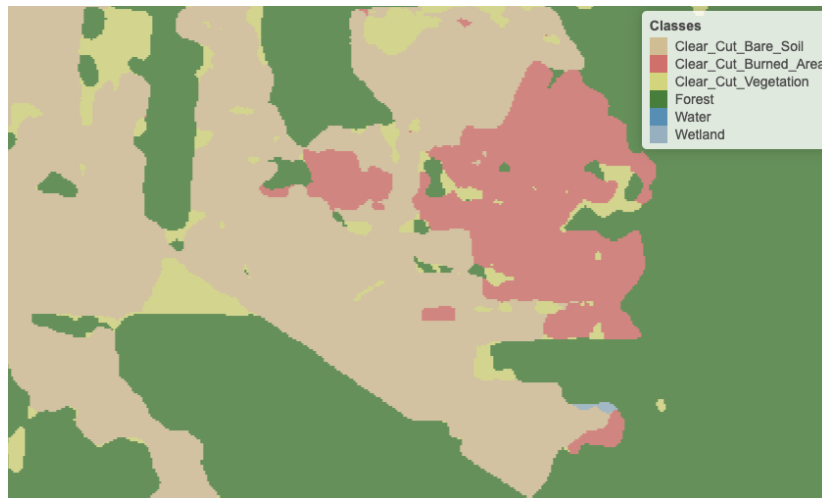


Figure 3: Detail of labelled map produced by pixel-based random forest after smoothing (source: authors)

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. The classification algorithm outputs the feature-based probability estimates $(p_{i,1}, \dots, p_{i,m})$ from (1). Let $\pi_{i,k} \geq 0$ be the prior probability of the i -th pixel belonging to class $k \in \{1, \dots, m\}$. We convert the observed values to the logit scale to allow for less modelling restrictions.

We assume that the logit of the prior probability of the pixels i associated to class k is described by a Gaussian distribution function

$$x_{i,k} = \log\left(\frac{\pi_{i,k}}{1 - \pi_{i,k}}\right) \sim N(m_{i,k}, s_{i,k}^2) \quad (2)$$

where $m_{i,k}$ represents the mean value and $s_{i,k}^2$ the class variance. We express the likelihood as a conditional Gaussian distribution of the logit $x_{i,k}$ of the observed values $p_{i,k}$ over $\mu_{i,k}$:

$$(x_{i,k} | \mu_{i,k}) = \log(p_{i,k} / (1 - p_{i,k})) \sim N(\mu_{i,k}, \sigma^2). \quad (3)$$

In the above equation, $\mu_{i,k}$ is the posterior expected mean of the logit probability associated to the i -th pixel. The variance σ_k^2 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

$$(\mu_{i,k} | x_{i,k}) \sim \sum N\left(\frac{m_{i,k}\sigma_k^2 + x_{i,k}s_{i,k}^2}{\sigma_k^2 + s_{i,k}^2}, \left(\frac{1}{\sigma_k^2} + \frac{1}{s_{i,k}^2}\right)^{-1}\right) \quad (4)$$

The estimate calculates the posterior distributions for each class; the pixel will be assigned to the class with higher posterior mean. The posterior mean is a weighted average between the observed logit value $x_{i,k}$ and the prior mean $m_{i,k}$. When the prior variance $s_{i,k}^2$ is high, the algorithm assigns more weight to the observed value $x_{i,k}$. Conversely, as the likelihood variance σ_k^2 increases, the update assigns more weight to the prior mean $m_{i,k}$.

Bayesian smoothing for land classification assumes that image patches with similar characteristics have a dominant class. This dominant class has higher average probabilities and lower variance than other classes. A pixel assigned to a different class will likely exhibit high 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 $s_{i,k}^2$. 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 neighbourhood, as explained below.

2.3. Empirical Bayes statistics with anisotropic neighbourhoods

Classification challenges arise for pixels located along the boundaries between areas containing different classes, as they possess signatures of two classes. In these cases, only some of the neighbours of such boundary pixels belong to the same class. To address this issue, we employ a non-isotropic definition of a neighbourhood to estimate the prior class distribution.

Consider a boundary pixel with a neighbourhood 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 a specific neighbourhoods $\mathcal{N}_{i,k}$ for each class k and pixel i . We use an L -statistic to estimate $m_{i,k}$ and $s_{i,k}^2$ 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

$$\hat{m}_{i,k} = \frac{1}{1-\alpha} \int_{\alpha}^{\infty} \mathbb{F}_{i,k}^{-1}(s) ds, \quad (5)$$

the average of the largest $(1-\alpha)$ -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 $\hat{s}_{i,k}^2$. The values of $\hat{m}_{i,k}$ and $\hat{s}_{i,k}^2$ are used in the Bayesian updating (cf Equation 4).

2.4. Effect of the hyperparameter

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

Thus, the parameter σ_k^2 expresses confidence in the inherent variability of the distribution of values of a class k . The smaller the parameter σ_k^2 , the more we trust the estimated values produced by the classifier for class k . Conversely, higher values of σ_k^2 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 a typical situation of 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 have confidence on the observed values, we can set σ^2 to be 5 for both classes A and B. In this case, the Bayesian probability estimate will be 0.48 for class A and 0.52 for class B. The original class will be kept. This example shows how the result of the Bayesian estimate 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 **bayesEO** contains functions that allow reading a probability file, applying Bayesian smoothing, and plotting the result. It also includes auxiliary functions for computing the variance of the probability maps.

```
R> # if bayesEO is not installed, install it
R> if (!requireNamespace("bayesEO", quietly = TRUE))
+   install.packages("bayesEO", dependencies = TRUE)
R> library(bayesEO)
```

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 what follows, 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 to 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/probs/", package = "bayesE0")
R> probs_file <- paste0(data_dir,
+                        "SENTINEL-2_MSI_20LLQ_2020-06-04_2021-08-26_probs_v0.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"))

```

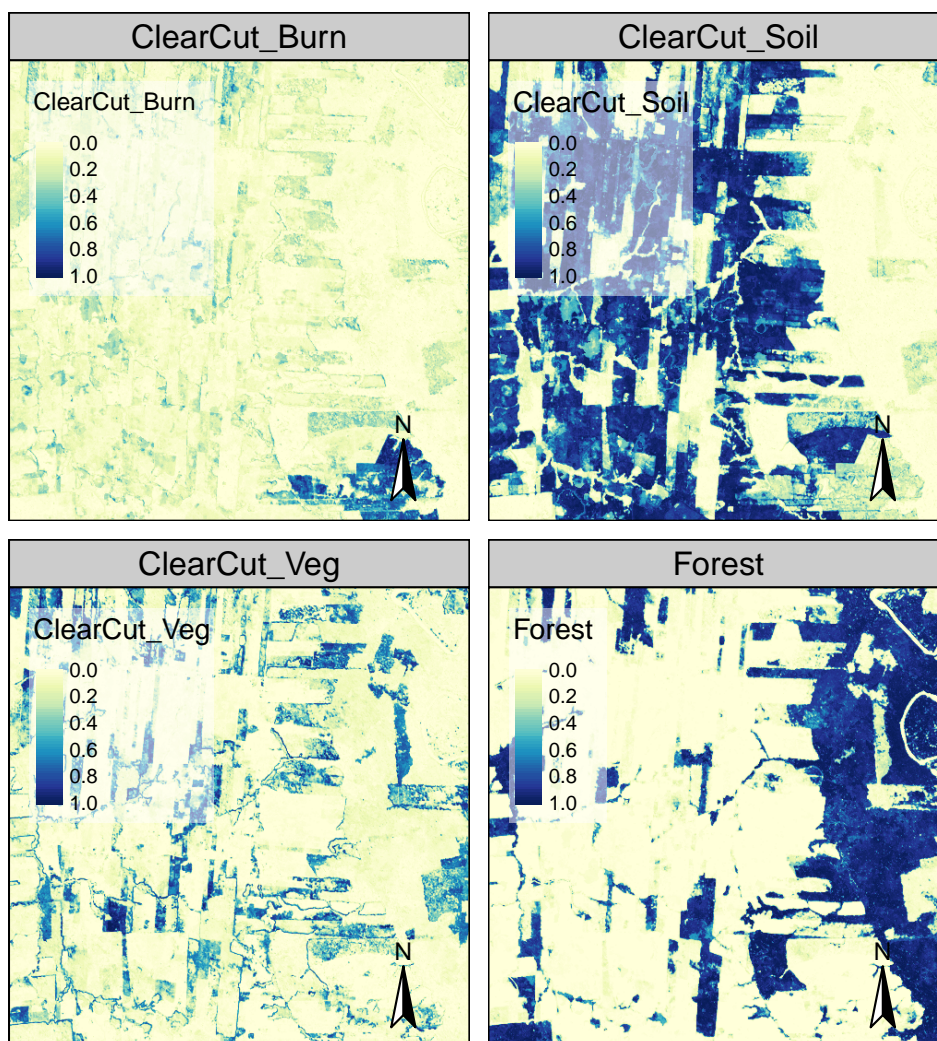


Figure 4: Class probabilities produced by random forest algorithm

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. Class ClearCut_Soil is mostly composed of dense areas of high probability whose geometrical boundaries result from forest cuts. 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_Burn include both homogeneous areas of high probability and areas of mixed response. Since classes have different behaviours, the post-processing procedure should enable users to control how to handle outliers and border pixels of each class.

The next step is to show the labelled map resulting from the raw class probabilities. Figure 5 shows the classification map, obtained by taking the class of higher probability to each pixel, without considering the spatial context. There are many places with the so-called “salt-and-pepper” effect which result from misclassified pixels. The non-smoothed labelled map shows the need for post-processing, since it contains a significant number of outliers and areas with mixed labelling.

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

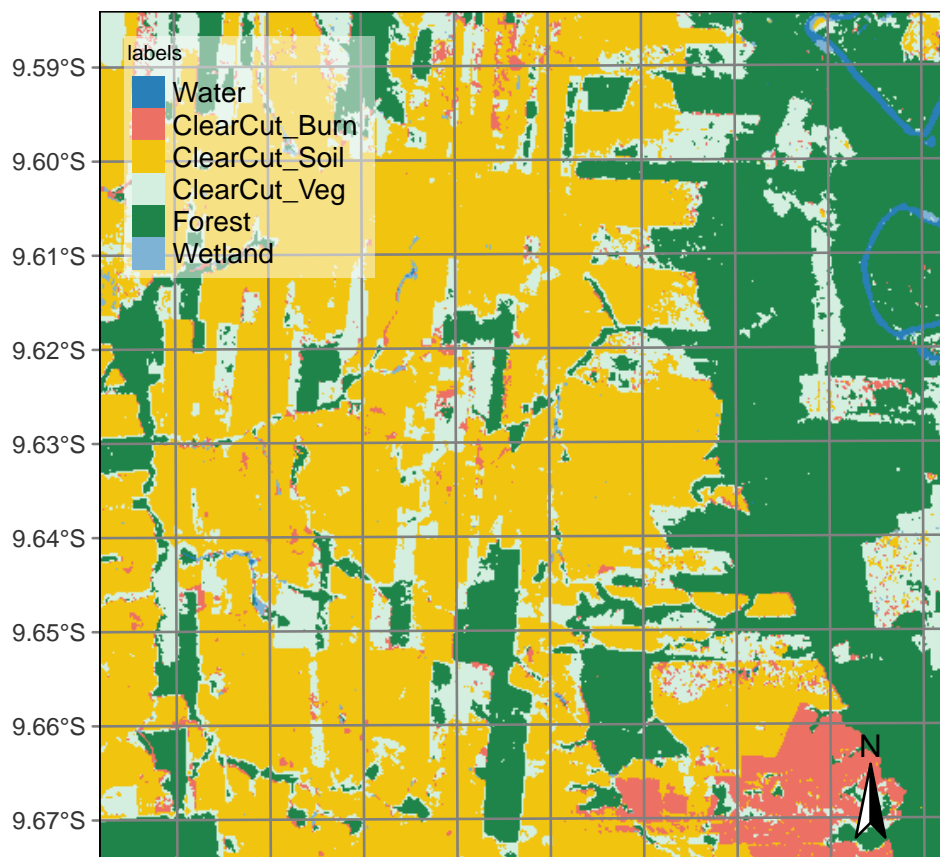


Figure 5: Labelled map without smoothing.

3.3. Estimating the local logit variances

The local logit variances correspond to the $s_{i,k}^2$ 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. : 0.002481	Min. : 0.003603	Min. : 0.0000
1st Qu.: 0.2017	1st Qu.: 0.072208	1st Qu.: 0.100210	1st Qu.: 0.1116
Median : 2.3265	Median : 0.150143	Median : 0.225121	Median : 0.2836
Mean : 2.6266	Mean : 0.331228	Mean : 0.614570	Mean : 0.6481
3rd Qu.: 4.7932	3rd Qu.: 0.310010	3rd Qu.: 0.550980	3rd Qu.: 0.7507
Max. : 21.7559	Max. : 11.260116	Max. : 10.368152	Max. : 16.8963
Forest	Wetland		
Min. : 0.0000	Min. : 0.000000		
1st Qu.: 0.1159	1st Qu.: 0.07297		
Median : 0.4922	Median : 0.16622		
Mean : 1.6533	Mean : 0.49490		
3rd Qu.: 2.5643	3rd Qu.: 0.35359		
Max. : 23.2210	Max. : 10.39689		

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`, `ClearCut_Burn`, `ClearCut_Veg`, and `Forest`. 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_Soil", "Forest"))
```

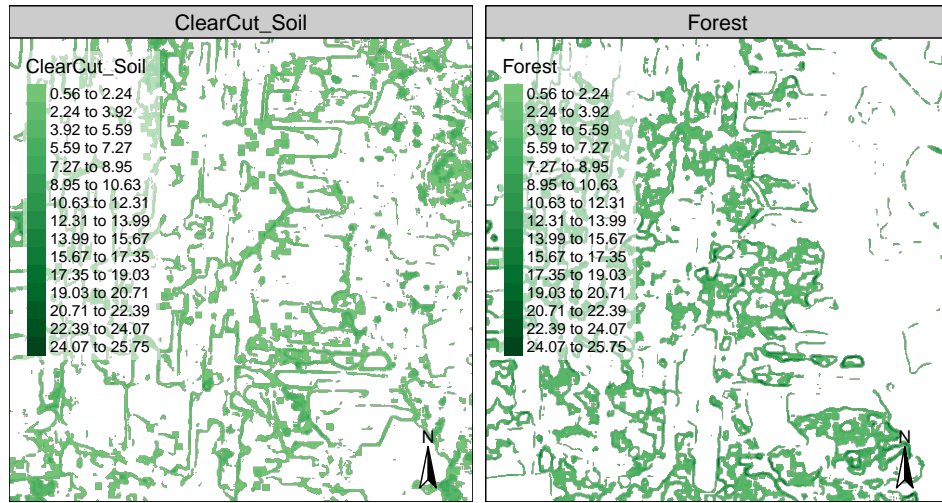


Figure 6: Logit variance map showing values above the 3rd quartile for classes ClearCutSoil and Forest.

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

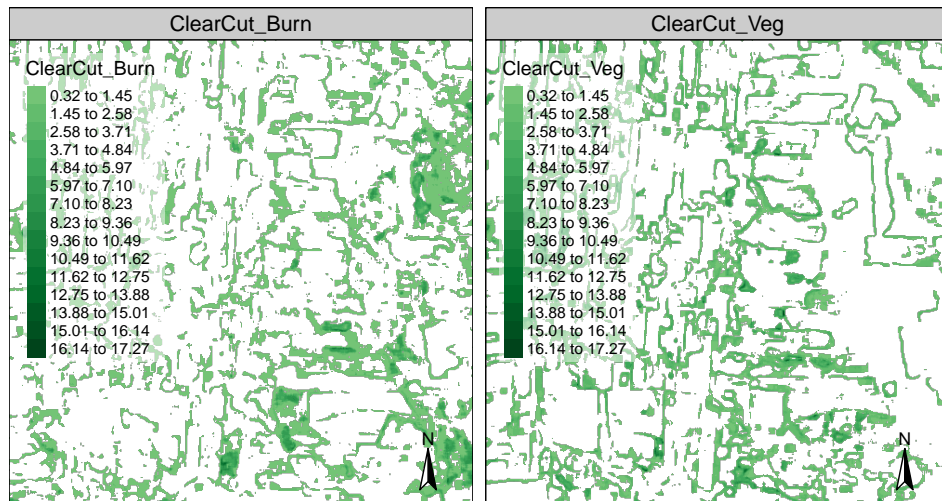


Figure 7: Logit variance map showing values above the 3rd quartile for classes ClearCutVeg and ClearCutBurn.

Comparing the variance maps of Figure 6 with the probability maps of Figure 4, one sees that areas of high probability of classes Forest and ClearCut_Soil are mostly made of compact patches. Recall these are the two dominant classes in the area, and deforestation is a process that converts forest to bare soil. Many areas of high logit variance for these classes are related to border pixels which have a mixed response. Areas of large patches of high logit variance for these classes are associated to lower class probabilities and will not be relevant to the final result.

By contrast, the transitional classes `ClearCut_Veg` and `ClearCut_Burn` have a different spatial pattern of their probability and logit variance. Class `ClearCut_Veg` has a high spatial variability, since pixels of this class arise when the forest has not been completely removed and there is some remaining vegetation after trees are cut. The extent of remaining vegetation after most trees have been removed is not uniform. For this reason, many areas of high local logit variance of class `ClearCut_Veg` are located in mixed patches inside pixels of class `Forest`.

Instances of class `ClearCut_Burn` arise following a forest fire. As shown in Figure 4, most pixels of this class tend to form mid-sized to large spatial clusters, because of how forest fires start and propagate. It is desirable to preserve the contiguity of the burned areas and remove pixels of other classes inside these clusters. Isolated points of class `ClearCut_Burn` can be removed without significant information loss.

What emerges from the above analysis is that each land cover and land use class has different expected spatial patterns. For this reason, it is useful to use the variance data to support adequate assignment of σ_k^2 value. We compute the top 25% of the logit variance values and use them for an informed choice of the hyperparameter.

```
R> # function to return quantiles for the logit variances
R> var_quantiles <- function(var_image, intervals, quantiles){
+   values <- terra::spatSample(var_image, size = 15000, na.rm = TRUE)
+   mat <- apply(values, 2, function(x){
+     quantile(x, probs = seq(0, 1, intervals))
+   })
+   return(mat[quantiles, ])
+ }
R> # get the 75%-95% quantiles
R> var_quantiles(var_image, 0.05, c("75%", "80%", "85%", "90%", "95%", "100%"))
```

	Water	ClearCut_Burn	ClearCut_Soil	ClearCut_Veg	Forest	Wetland
75%	4.764197	0.3147544	0.5568953	0.7583870	2.662210	0.3547823
80%	4.971079	0.3789011	0.7222972	0.9521859	3.433133	0.4273479
85%	5.148950	0.4670305	1.0232807	1.2087787	4.324530	0.5382029
90%	5.422300	0.6047362	1.5498801	1.6012467	5.040908	0.7697119
95%	6.016208	0.9556933	2.8438386	2.3973728	5.835870	2.3830680
100%	24.353707	12.8347503	14.2305099	14.4590720	21.898716	11.3679790

3.4. Applying Bayesian smoothing to remove outliers

To remove the outliers, **bayesEO** provides `bayes_smooth()`. Its main parameters are: (a) `x`, a probability image; (b) `window_size`, dimension of the local neighbourhood; (c) `smoothness`, values of σ_k^2 for each class. The impact of Bayesian smoothing can be best captured by producing a labelled map using `bayes_label()`, taking the smoothed image as its input.

We choose values of σ_k^2 that reflect our prior expectation of the spatial patterns of each class. For classes `ClearCut_Veg` and `ClearCut_Burn`, to produce denser spatial clusters and remove “salt-and-pepper” outliers, we take σ_k^2 values in 95%-100% range. In the case of the most frequent classes `Forest` and `ClearCut_Soil` we want to preserve their original spatial shapes as much as

possible; the same logic applies to less frequent classes Water and Wetlands. For this reason, we set σ_k^2 values in the 75%-80% range for these classes. Figure 8 shows that the outliers and isolated pixels have been removed. The class spatial patterns correspond to our prior expectations.

```
R> map_smooth_bayes <- bayes_smooth(probs_image,
+   window_size = 7,
+   smoothness = c( "Water" = 5.0, "ClearCut_Burn" = 20, "ClearCut_Soil" = 1,
+   "ClearCut_Veg" = 15, "Forest" = 3.5, "Wetland" = 0.40)
+ ) |> bayes_label()
R> bayes_plot_map(map_smooth_bayes)
```

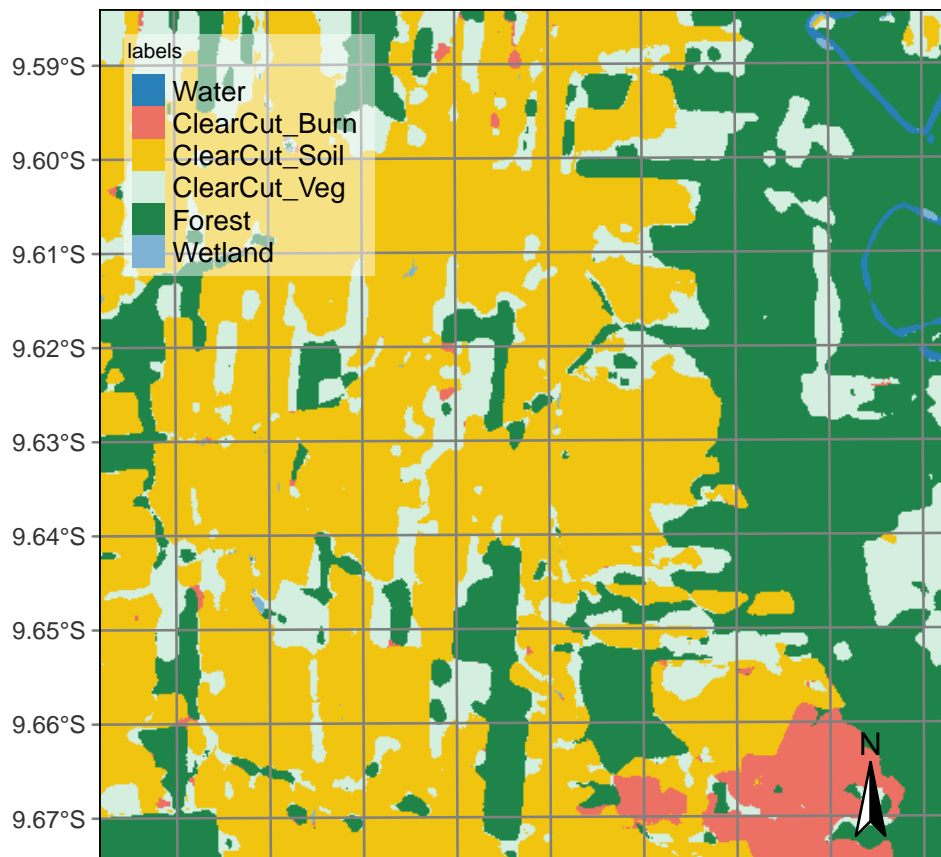


Figure 8: Labeled map with Bayesian smoothing.

In the smoothed map (Figure 8), the most frequent classes (ClearCut_Soil and Forest) increased slightly their areas at the expense of the others. The class ClearCut_Burn now appear as dense patches as befits areas affected by forest fire, while class ClearCut_Veg has preserved most of their areas, while removing outliers.

4. Comparison with other methods

To better understand the benefits of Bayesian smoothing, 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.

$$I'(x, y) = \sum_{i=-k}^k \sum_{j=-k}^k \frac{1}{2\pi\sigma^2} e^{-\frac{(x-i)^2 + (y-j)^2}{2\sigma^2}} \cdot I(x-i, y-j) \quad (6)$$

In this equation, $I'(x, y)$ represents the smoothed image, $I(x, y)$ is the original image, (i, j) are the indices running over the kernel size, and where σ is the standard deviation a Gaussian distribution. The following code computes a Gaussian filter, labels the image and plots the resulting map.

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

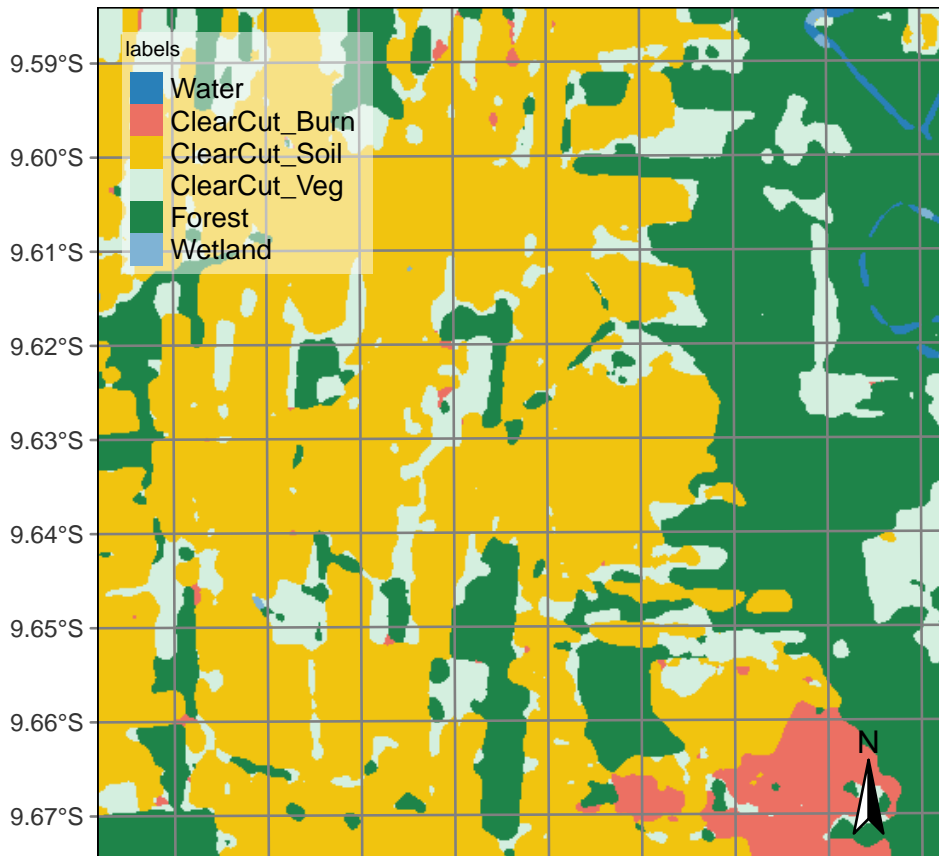


Figure 9: Labeled map using Gaussian smoothing.

Comparison between Figure 8 and Figure 9 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 for each class. 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

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

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

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

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

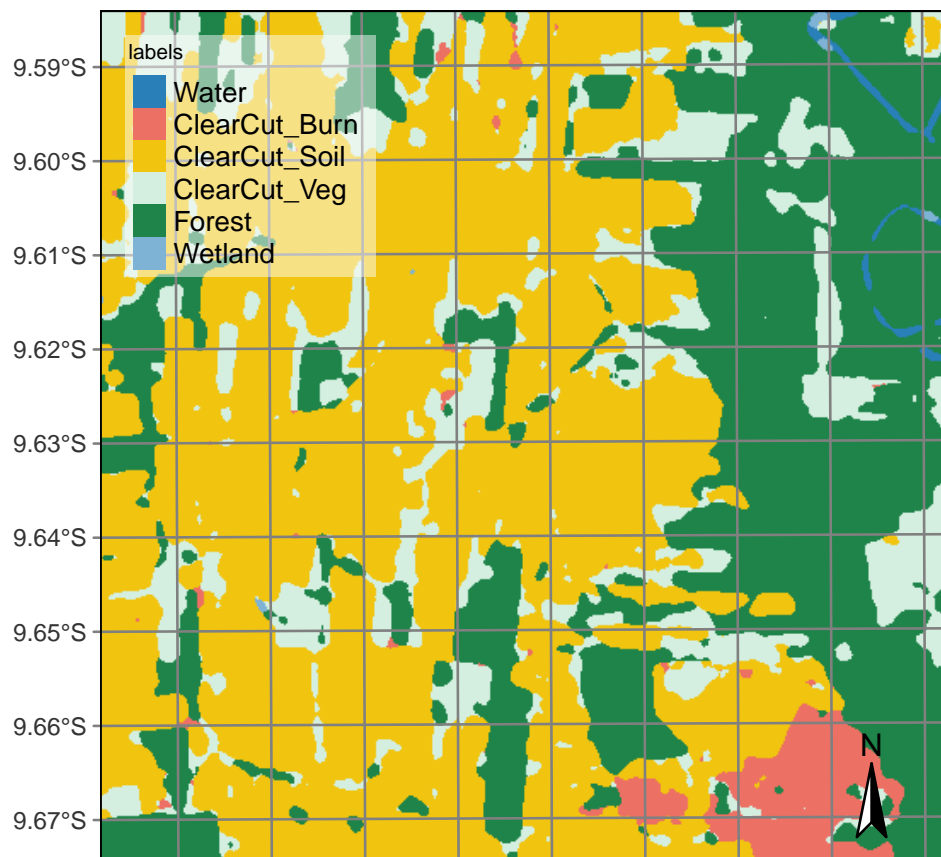


Figure 10: Labeled map using bilateral smoothing.

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 based on expert knowledge. To better assert the impact of smoothing, consider the class areas in the non-smoothed and smoothed maps and compare with the results of gaussian and bilateral filters.

```
R> # get the summary of the non-smoothed map
R> sum1 <- bayes_summary(map_no_smooth)
R> colnames(sum1) <- c("class", "no_smooth")
R> # get the summary of the smoothed map
R> sum2 <- bayes_summary(map_smooth_bayes)
R> colnames(sum2) <- c("class", "smooth")
R> # get the summary of the gaussian map
R> sum3 <- bayes_summary(map_gaussian)
R> colnames(sum3) <- c("class", "gauss")
R> # get the summary of the bilateral map
R> sum4 <- bayes_summary(map_bilat)
R> colnames(sum4) <- c("class", "bilat")
R> # compare class areas of non-smoothed and smoothed maps
```

```
R> dplyr::inner_join(sum1, sum2, by = "class") |>
+   dplyr::inner_join(sum3, by = "class") |>
+   dplyr::inner_join(sum4, by = "class")
```

```
# A tibble: 6 x 5
  class      no_smooth smooth  gauss  bilat
  <chr>      <dbl>  <dbl>  <dbl>  <dbl>
1 Water      0.51   0.46   0.38   0.4
2 ClearCut_Burn 4.59   3.19   3.08   3.08
3 ClearCut_Soil 47.2   48.9   51.0   50.9
4 ClearCut_Veg 17.1   16.0   14.1   14.2
5 Forest     30.2   31.3   31.4   31.3
6 Wetland     0.38   0.13   0.087  0.088
```

Bayesian smoothing does a better job at preserving the original class areas, while producing compact shapes and removing outliers. While the area differences in this example are relatively small, they tend to increase in large regions with multiple classes. This example shows the value of the Bayesian inference procedure compared with Gaussian and bilateral filtering. 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. 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. Recent reviews of deep learning classification using image time series highlight the value of post-classification spatial smoothing ([Perbet *et al.* 2024](#)). Thus, Bayesian smoothing is an important component of the time-first, space-later Earth observation data analysis.

The Bayesian smoothing method proposed in the paper has proven valuable in a number of studies that use the R **sits** for land use classification of image time series (Picoli *et al.* 2018; Simoes *et al.* 2020; Hadi *et al.* 2023; Giuliani 2024; Werner *et al.* 2024). The availability of the method as a stand-alone **bayesEO** is intended to broaden the target user community and make it easier for R experts to combine the algorithm in their scripts.

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 available 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 DE-TER 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 Roger EP (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, Burlingame, 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.

- Giuliani G (2024). “Time-First Approach for Land Cover Mapping Using Big Earth Observation Data Time-Series in a Data Cube – a Case Study from the Lake Geneva Region (Switzerland).” *Big Earth Data*, **0**(0), 1–32. ISSN 2096-4471. doi:[10.1080/20964471.2024.2323241](https://doi.org/10.1080/20964471.2024.2323241).
- 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](https://doi.org/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](https://doi.org/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](https://doi.org/10.1016/j.isprsjprs.2019.04.015).
- Perbet P, Guindon L, Côté JF, Béland M (2024). “Evaluating Deep Learning Methods Applied to Landsat Time Series Subsequences to Detect and Classify Boreal Forest Disturbances Events: The Challenge of Partial and Progressive Disturbances.” *Remote Sensing of Environment*, **306**, 114107. ISSN 0034-4257. doi:[10.1016/j.rse.2024.114107](https://doi.org/10.1016/j.rse.2024.114107).
- 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](https://doi.org/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](https://doi.org/10.3390/rs13132428).
- Simoes R, Picoli MCA, Camara G, Maciel A, Santos L, Andrade PR, Sánchez A, Ferreira K, Carvalho A (2020). “Land Use and Cover Maps for Mato Grosso State in Brazil from 2001 to 2017.” *Scientific Data*, **7**(1), 34. ISSN 2052-4463. doi:[10.1038/s41597-020-0371-4](https://doi.org/10.1038/s41597-020-0371-4).
- 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](https://doi.org/10.1109/ICCV.1998.710815).

Werner JPS, Belgiu M, Bueno IT, Dos Reis AA, Toro APSGD, Antunes JFG, Stein A, Lamparelli RAC, Magalhães PSG, Coutinho AC, Esquerdo JCDM, Figueiredo GKDA (2024). “Mapping Integrated Crop–Livestock Systems Using Fused Sentinel-2 and PlanetScope Time Series and Deep Learning.” *Remote Sensing*, **16**(8), 1421. ISSN 2072-4292. doi:[10.3390/rs16081421](https://doi.org/10.3390/rs16081421).

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](https://doi.org/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