



## **Bayesian inference for smoothing of remote sensing image classification using machine learning**

**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**

The abstract of the article.

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

---

## 1. Introduction

Machine learning methods such as support vector machines (Mountrakis *et al.* 2011), random forests (Belgiu and Dragut 2016), and deep learning (Ma *et al.* 2019) have become indispensable in the field of remote sensing image classification. These techniques typically rely on training samples that are derived from “pure” pixels, handpicked by users to represent the desired output classes. However, considering the presence of mixed pixels in images regardless of resolution, and the considerable data variability within each class, these classifiers often produce results with outliers or misclassified pixels. Therefore, post-processing techniques have become crucial to refine the labels of a classified image, thereby enhancing its accuracy (Huang *et al.* 2014).

Post-processing is a necessary step in remote sensing image classification. They reduce the salt-and-pepper effect, where single pixels or small groups of pixels are classified differently from their larger surrounding areas; this effect leads to visual discontinuity and inconsistency. By mitigating these errors and minimizing noise, post-processing improves the quality of the initial classification results. There is a significant gain in the overall accuracy and interpretability of the final output (Schindler 2012).

Most post-processing methods use the smoothness assumption that nearby pixels tend to have the same label (Schindler 2012). These include probability-based smoothing methods such as Gaussian and edge-aware filtering (Schindler 2012), modal filters (Ghimire *et al.* 2010), and probabilistic relaxation (Gong and Howarth 1989). The method proposed by Huang *et al.* (2014) involves a relearning method based on co-occurrence matrices, which represent the joint distribution of class labels in the vicinity of each pixel. In this paper, we propose a post-processing algorithm founded on Bayesian statistics.

Most post-processing methods operate under the “smoothness assumption” that nearby pixels are likely to have the same label (Schindler 2012). These include probability-based smoothing methods such as Gaussian and edge-aware filtering (Schindler 2012), modal filters (Ghimire *et al.* 2010), and probabilistic relaxation (Gong and Howarth 1989). The method proposed by Huang *et al.* (2014) involves a relearning method based on co-occurrence matrices, which represent the joint distribution of class labels in the vicinity of each pixel. In contrast, we propose a post-processing algorithm founded on Bayesian statistics.

Bayesian inference is a way of coherently update our uncertainty in the light of new evidence. It can consider expert knowledge on the derivation of probabilities. In Spatial Statistics, Bayesian inference has been used in many problems, including the analysis of remote sensing images expressed as lattice data. Lattice processes  $\{Y(\mathbf{s}) : \mathbf{s} \in D_s\}$  are defined on a finite and countable subset  $D_s$  of  $\mathbb{R}^d$  (Cressie and Wikle 2011), consisting of discrete spatial locations such as pixels or administrative units. The standard model for lattice data is a spatial process  $Z \equiv (Z(\mathbf{s}_1), \dots, Z(\mathbf{s}_n))$ , where  $S = \mathbf{s}_1, \dots, \mathbf{s}_n$  are the locations. These spatial random process are associated to a single continuous variable, e.g., temperature distribution on the Earth’s surface. While assumptions of spatial continuity and dependence hold true for all locations within the lattice space for continuous variables, they cannot be extended to land classification scenarios.

The problem of land classification can be expressed as follows. Given a set of spatial locations  $S = \mathbf{s}_1, \dots, \mathbf{s}_n$  and a set of land classes  $L = l_1, \dots, l_k$ , we seek a classification function such that

$$f : S \rightarrow L, \forall s \in S, f(s) = (p_1(l_1), \dots, p_k(l_k)), \sum_{(i=1, \dots, k)} p_i(l_i) = 1 \quad (1)$$

For each pixel, the classification function produces a set of class probabilities that sum to 100%. The function  $f$  is obtained from a training dataset  $T = (s_1, l_{s1}), \dots, (s_m, l_{sm})$  where for a subset  $S_m \in S$  we have access to one label per location. In the general case, the subset  $S_m$  is much smaller than  $S$ . The unique nature of land classification, compared to other forms of spatial analysis, lies in the behavior of the random processes linked to the spatial locations. Land classification aims to subdivide space into discrete areas, each associated with a distinct type of land use or cover. Consequently, the random processes correlated with each class are not spatially continuous. Take, for instance, Figure 2 (fig:roim) showcasing a fragment of a Sentinel-2 image in the state of Rondonia, Brazil. The image starkly contrasts forest areas (depicted in green) and deforested areas (portrayed in various shades of orange and brown). In terms of spatial processes, there's a discontinuity at the borders separating forested and deforested regions. Therefore, in land classification we need to make different assumptions about the statistical properties of the data. These considerations serve to capture the unique spatial patterns and heterogeneity inherent in land classifications, which traditional spatial processes might not adequately represent.



Figure 1: Sentinel-2 image of deforestation in the Amazon forest (detail).

When modelling a multi-class land classification as a set of spatial processes,  $Z_i \equiv (Z_i(\mathbf{s}_1), \dots, Z_i(\mathbf{s}_n))$ , where  $S = \mathbf{s}_1, \dots, \mathbf{s}_n$  are the locations and where  $i \in 1, \dots, k$  represents the land classes, we cannot assume that Tobler's First Law of Geography ("*everything is related to everything else, but near things are more related than distant things.*") holds between all spatial locations. Land classification produces results where spatial classes are associated to compact patches. Inside the patches, one can assume the existence of a random spatial process whose realization supports the properties of continuity and spatial correlation. However, such properties break at the borders between land classes.

The application of Bayesian inference for the smoothing of land classification images, produced by machine learning techniques needs to make additional assumptions than traditional methods used in spatial statistics. We need to take into account discontinuities at the borders between

land classes. Based on this considerations, in this paper we introduce the `bayesEO` package. This package presents a novel method for image post-processing, grounded in Bayesian statistics, which caters to the specific properties of land classification. We employ non-isotropic neighbourhood definitions to mimic the effects of discontinuity between land classes, enabling our method to better capture the spatial complexities inherent in such data. Our method allows the inclusion of expert knowledge, removes outliers, and enhances the consistency of the resulting map. To the best of our knowledge, this work represents the first proposal to utilize Bayesian inference for the post-processing of machine learning-based image classification. This innovation stands to substantially improve the accuracy and interpretability of land classification results.

In R, ... include here a discussion of what's available in R, and include mention to the `sits` package.

## 2. Methods

### 2.1. Conversion from probabilities to logits

The proposed post-classification smoothing model considers the output of a machine learning algorithm that provides the probabilities of each pixel in the image to belong to target classes. More formally, let  $p_{i,k} \geq 0$  be the probability of pixel  $i$  belonging to class  $k$ ,  $k = 1, \dots, K$ , where  $K$  is the number of classes, then

$$\sum_{k=1}^K p_{i,k} = 1. \quad (2)$$

To make the inference tractable, the class probability values  $p_{i,k}$  are converted to log-odds values. The logit function converts probability values in the interval  $[0, 1]$  to logit values in  $[-\infty, \infty]$ , expressed by

$$x_{i,k} = \ln\left(\frac{p_{i,k}}{1 - p_{i,k}}\right). \quad (3)$$

Confidence in pixel classification increases with logit. There are situations, such as border pixels or mixed ones, where the logit of different classes are similar in magnitude. These are cases of low confidence in the classification result. To assess and correct these cases, the post-classification smoothing method borrows strength from the neighbors.

### 2.2. Bayesian inference

Once the probability values are converted into logits, we can frame the problem within a Bayesian context. As discussed in our earlier section on statistical models for land classification, each land class (e.g., "Forest", "Deforestation", "Grasslands") is linked to a distinct spatial process. Thus, the number of spatial processes equals the number of classes, and each process exhibits unique behaviours distinct from the others. This diversity in class behavior leads us to posit independence between the classes. Thus, for each spatial location, the model uses the probability values associated to each class as realization of a specific spatial process. This perspective allows us to account for the intricate spatial patterns and variations inherent to each land class in our Bayesian analysis.

We will then consider each pixel  $i$  and class  $k$  to be associated to a random variable  $Z_{i,k}$ . Given the probability value  $x_{i,k}$  of class  $k$  at pixel  $i$  produced by the machine learning algorithm, we aim to estimate  $E[\mu_{i,k}|x_{i,k}]$ , the expected mean logit values which would correspond to the ground truth. The Bayesian inference procedure can be expressed as

$$\pi(\mu|x) \propto \pi(x|\mu)\pi(\mu). \quad (4)$$

To estimate the conditional posterior distribution  $\pi(\mu|x)$ , we combine two distributions: (a) the distribution  $\pi(x|\mu)$ , known as the likelihood, which expresses how measured values  $x_{i,k}$  depend in the unknown values  $\mu_{i,k}$ ; and (b)  $\pi(\mu)$ , which is our guess on the actual data distribution, known as the prior. The update will be performed for each class  $k$  separately.

The method assumes that the likelihood  $x_{i,k}|\mu_{i,k}$  follows a normal distribution  $N(\mu_{i,k}, \sigma_k^2)$ , with parameters  $\mu_{i,k}$  and  $\sigma_k^2$ . The variance  $\sigma_k^2$  will be estimated based on user expertise and taken as a hyperparameter to control the smoothness of the resulting estimate.

We will also assume a normal local prior for the parameter  $\mu_{i,k}$  with parameters  $m_{i,k}$  and  $s_{i,k}^2$ :

$$\mu_{i,k} \sim N(m_{i,k}, s_{i,k}^2). \quad (5)$$

The method assumes that the class probabilities in the spatial neighbourhood of the pixel also follow a normal distribution. Estimating the local means and variances for the prior distribution considers a spatial neighbourhood  $N_i$  close to pixel  $p_i$ . Let  $\#(N_i)$  be the number of elements in the neighbourhood  $N_i$ . The mean value is given by

$$m_{i,k} = \frac{\sum_{(j) \in N_i} x_{j,k}}{\#(N_i)} \quad (6)$$

and the variance by

$$s_{i,k}^2 = \frac{\sum_{(j) \in N_i} [x_{j,k} - m_{i,k}]^2}{\#(N_i) - 1}. \quad (7)$$

Given these assumptions, the Bayesian update for the expected conditional mean  $E[\mu_{i,k}|x_{i,k}]$  is given by:

$$E[\mu_{i,k}|x_{i,k}] = \frac{m_{i,k} \times \sigma_k^2 + x_{i,k} \times s_{i,k}^2}{\sigma_k^2 + s_{i,k}^2}, \quad (8)$$

which can be expressed as a weighted mean

$$E[\mu_{i,k}|x_{i,k}] = \left[ \frac{s_{i,k}^2}{\sigma_k^2 + s_{i,k}^2} \right] \times x_{i,k} + \left[ \frac{\sigma_k^2}{\sigma_k^2 + s_{i,k}^2} \right] \times m_{i,k}, \quad (9)$$

where:

- $x_{i,k}$  is the logit value for pixel  $i$  and class  $k$ .
- $m_{i,k}$  is the average of logit values for pixels of class  $k$  in the neighborhood of pixel  $i$ .

- $s_{i,k}^2$  is the variance of logit values for pixels of class  $k$  in the neighborhood of pixel  $i$ .
- $\sigma_k^2$  is the likelihood variance for class  $k$ , an user-derived hyperparameter.

The above equation is a weighted average between the value  $x_{i,k}$  for the pixel and the mean  $m_{i,k}$  for the neighboring pixels. When the variance  $s_{i,k}^2$  for the neighbors is too high, the algorithm gives more weight to the pixel value  $x_{i,k}$ . When the variance of the likelihood  $\sigma_k^2$  increases, the method gives more weight to the neighborhood mean  $m_{i,k}$ .

### **2.3. Effect of the hyperparameter**

The parameter  $\sigma_k^2$  controls the level of smoothness. If  $\sigma_k^2$  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  $\sigma_k^2$ , which are small relative to the prior variance  $s_{i,k}^2$ , increase our confidence in the original probabilities. Conversely, likelihood variances  $\sigma_k^2$ , which are large relative to the prior variance  $s_{i,k}^2$ , increase our confidence in the average probability of the neighborhood.

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

Consider the following two-class example. Take a pixel 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. Suppose a 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  $\sigma_k^2$ , representing prior belief in the variability of the probability values for each class. If we take both  $\sigma_A^2$  for class A and  $\sigma_B^2$  for class B to be both 10, the Bayesian estimated probability for class A is 0.52 and for class B is 0.48. In this case, the pixel will be relabeled as being class A. If we set  $\sigma^2$  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.

### **2.4. Defining non-isotropic neighbourhoods**

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

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.

The non-isotropic neighborhood definition is selected as a subset of the logits of all values in a user-defined window. For instance, 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 posterior 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 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.

### 3. Software and examples

#### 3.1. Reading a probability data cube

The input for post-classification is an image with probabilities produced by a machine learning algorithm. This file should be multi-band, where each band contains the pixel probabilities of a single class. The file name must have information on reference dates and include a version number. 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 image has been stored as INT2S data type with integer values between [0..10000] to represent probabilities ranging from 0 to 1.

```
R> data_dir <- system.file("/extdata/Rondonia-20LLQ/", package = "sitsdata")
R> probs_file <- list.files(data_dir)
R> probs_file
[1] "SENTINEL-2_MSI_20LLQ_2020-06-04_2021-08-26_probs_v1.tif"
```

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 removed; (f) ClearCut\_Veg where some vegetation remains after most trees have been removed. The class labels should also be informed by the user, since they are not stored in image files.

```
R> labels <- c("Water", "ClearCut_Burn", "ClearCut_Soil",
+           "ClearCut_Veg", "Forest", "Wetland")
```

The following code reads the file using the `terra` package.

```
R> probs_image <- terra::rast(paste0(data_dir, "/", probs_file))
R> names(probs_image) <- labels
```

The output is a `SpatRaster` object from the `terra` package. Figure 2 shows the plot of all layer of the probability image. The map for class Forest shows high probability values associated with compact patches and linear stretches in riparian areas. Class ClearCut\_Soil is mostly composed of dense areas of high probability whose geometrical boundaries result from forest cuts. By contrast, the probability maps for classes Water, ClearCut\_Burn, and ClearCut\_Veg have mostly low values. Note that we need to inform the scaling parameter that converts the image to [0..1] interval.

```
R> bayes_plot(probs_image, scale = 0.0001)
```

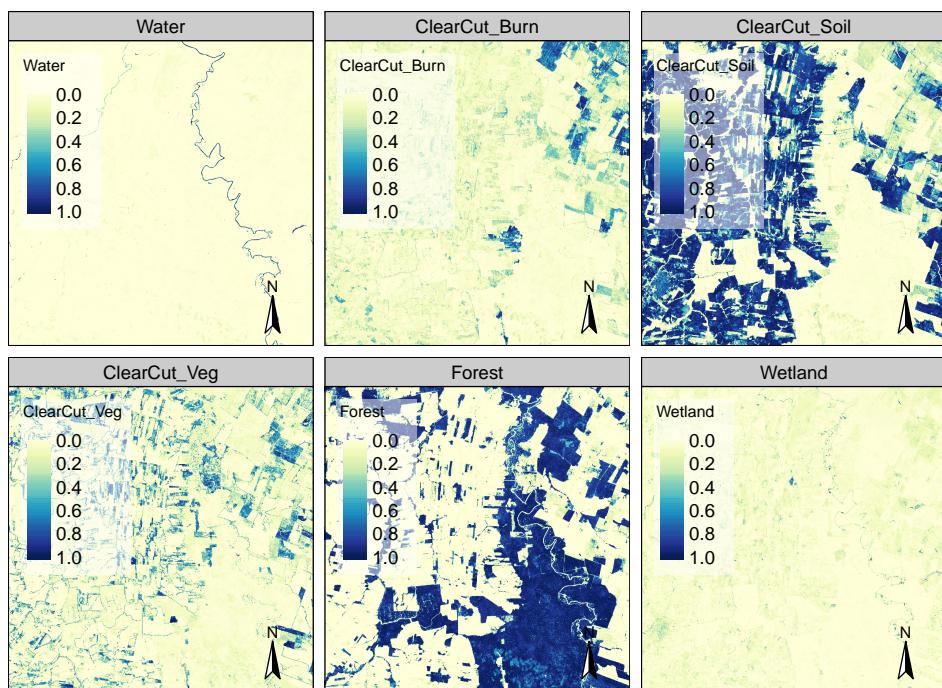


Figure 2: Class probabilities produced by random forest algorithm.

Figure 3 shows the resulting 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 misclassified pixels. The map is produced by `bayes_label()` whose parameter is a `SpatRaster` object containing the probabilities of each class for all pixels. Then the map is rendered using `bayes_map()`.

```
R> map_no_smooth <- bayes_label(probs_image)
```

```
R> bayes_map(map_no_smooth)
```

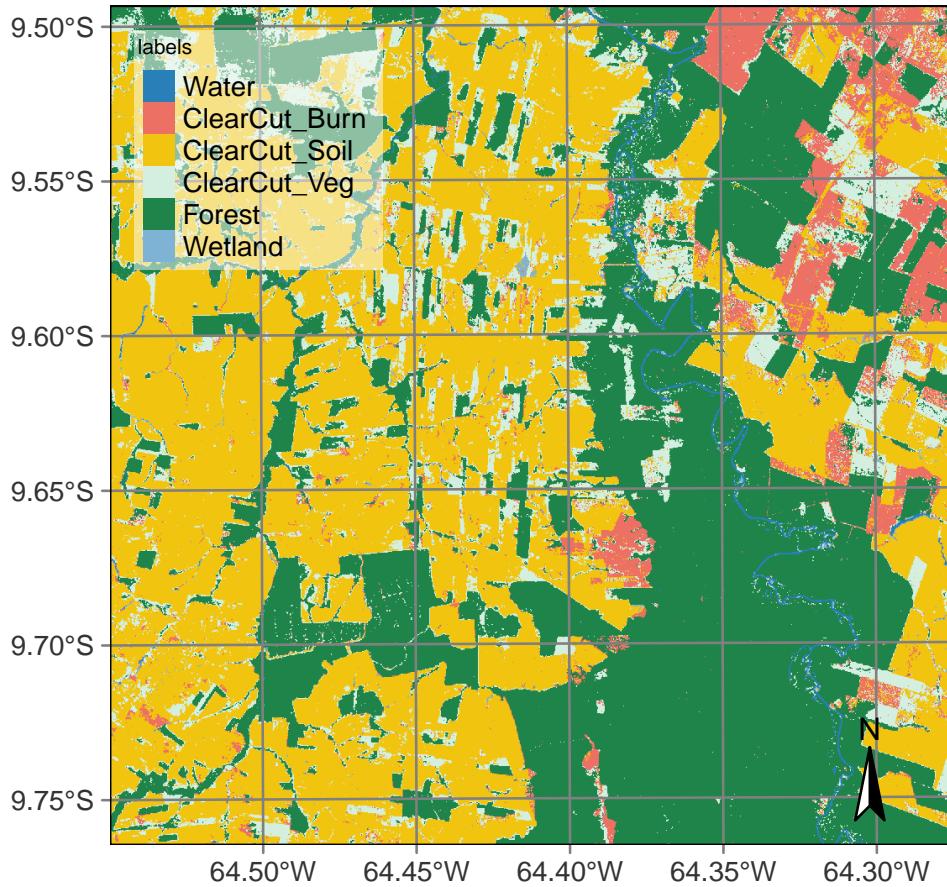


Figure 3: Labelled map without smoothing.

### 3.2. 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 `sits_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  $7 \times 7$  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)
R> bayes_summary(var_image)
```

Water	ClearCut_Burn	ClearCut_Soil	ClearCut_Veg
Min. : 0.0000	Min. : 0.00000	Min. : 0.00000	Min. : 0.00000
1st Qu.: 0.1612	1st Qu.: 0.06956	1st Qu.: 0.09601	1st Qu.: 0.09829
Median : 2.0058	Median : 0.16171	Median : 0.23713	Median : 0.24041
Mean : 2.4973	Mean : 0.43501	Mean : 0.86547	Mean : 0.68747

3rd Qu.: 4.5969	3rd Qu.: 0.35328	3rd Qu.: 0.65092	3rd Qu.: 0.70635
Max. :28.0902	Max. :14.15232	Max. :13.61211	Max. :16.33909
Forest	Wetland		
Min. : 0.0000	Min. : 0.00000		
1st Qu.: 0.1126	1st Qu.: 0.07133		
Median : 0.4552	Median : 0.17513		
Mean : 1.6048	Mean : 0.70143		
3rd Qu.: 2.3851	3rd Qu.: 0.42119		
Max. :25.1679	Max. :11.39766		

The choice of the  $7 \times 7$  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 4 shows the values of local logit variances for classes ClearCut\_Soil and Forest, considering only the 4<sup>th</sup> 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_image, quantile = 0.75, labels = c("ClearCut_Soil", "Forest"))
```

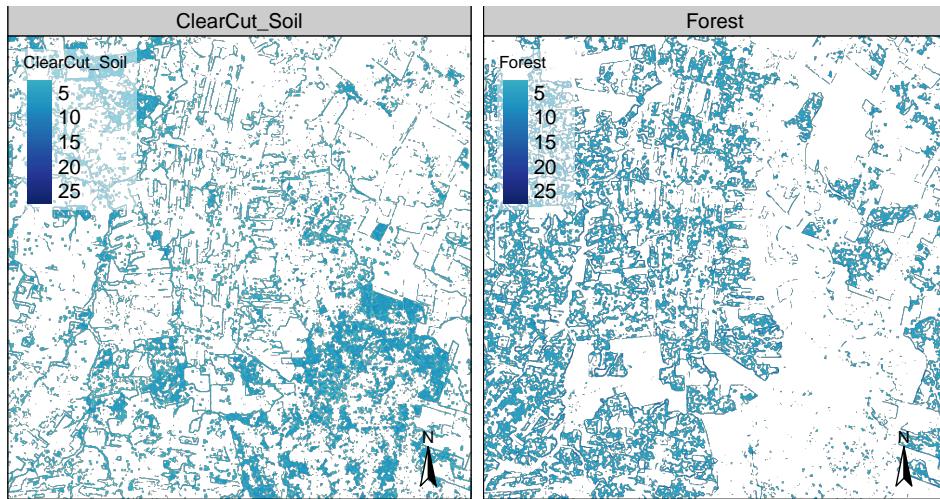


Figure 4: Logit variance map showing values above the 3rd quartile.

Comparing the logit variance maps of Figure 4 with the probability maps of Figure 2 emphasizes the relevance of expert knowledge. The areas of high probability of class Forest are mostly made of compact patches; areas of high local variance occur near the borders of these patches. By contrast, class ClearCut\_Veg represents a transition between natural forest areas and places where all trees have 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, the local variance of class `ClearCut_Veg` is mostly patch-based, while that of class `Forest` is mostly border-based.

Further insights are provided by Figure 5, which shows the histograms of local variances per class. The values shown correspond to the 4<sup>th</sup> quartile (top 25% of all values). The distribution of logit variances is uneven between the classes. Class `ClearCut_Veg` has a more balanced distribution, while most values in the 4<sup>th</sup> quartile of class `ClearCut_Soil` have low values.

```
R> bayes_hist(var_image, quantile = 0.75)
```

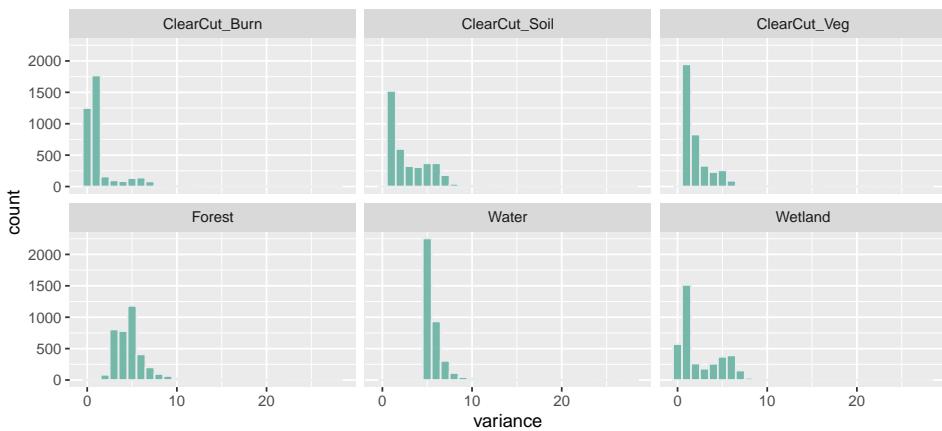


Figure 5: Histogram of top quartile of variances of class logits.

### 3.3. Applying Bayesian smoothing to remove outliers

As discussed above, the effect of the Bayesian estimator depends on the values of the a priori variance  $\sigma_k^2$  set by the user and the neighbourhood definition to compute the local variance  $s_{i,1}^2$  for each pixel. To show the effects of different  $\sigma^2$  values we consider two cases: (a) setting  $\sigma^2$  to a high value close to the maximum value of local logit variance; (b) setting  $\sigma^2$  to a lower value close to the minimum value of the 4<sup>th</sup> quartile. 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. We first consider the case of high  $\sigma^2$  values.

```
R> smooth_high <- bayes_smooth(
+   probs_image,
+   window_size = 7,
+   smoothness = c(25, 10, 20, 20, 25, 10)
+ )
```

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 6 shows that the outliers and isolated pixels have been removed.

```
R> map_smooth_high <- bayes_label(smooth_high)
R> bayes_map(map_smooth_high)
```

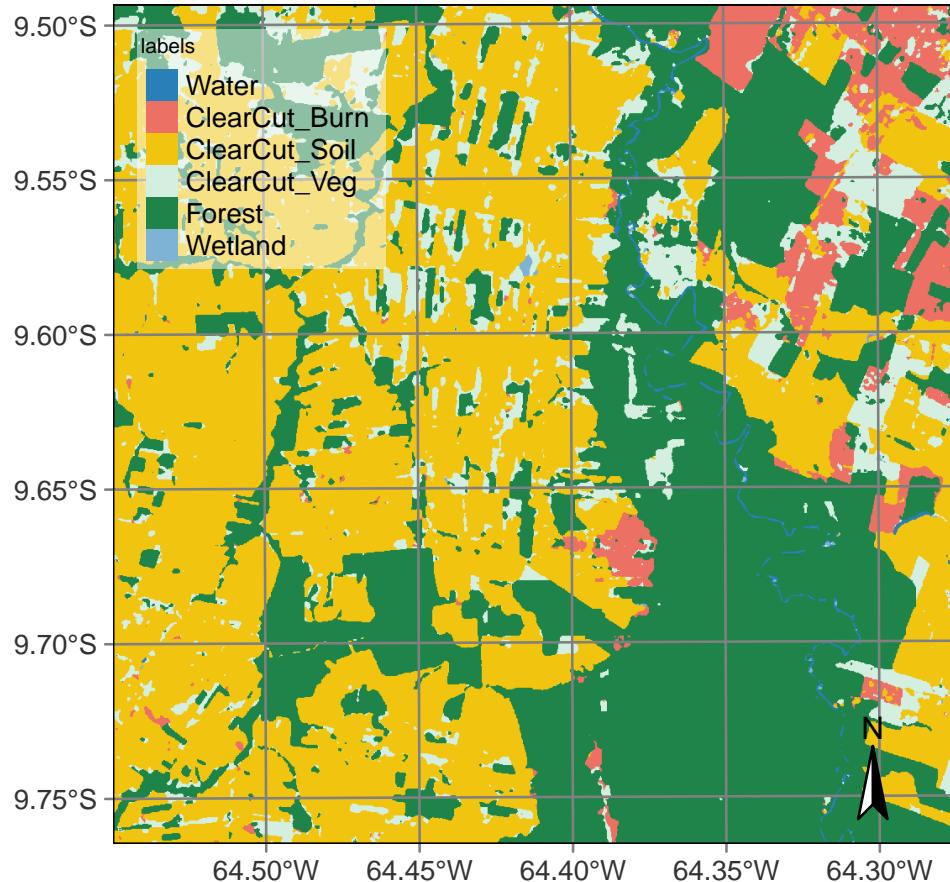


Figure 6: Labeled map with smoothing with high smoothness values.

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. Compare the smoothing with high values with the smoothing with values close to the minimum value of the 4<sup>th</sup> quartile of the local variance for each class, as computed below.

```
R> smooth_low <- bayes_smooth(
+   probs_image,
+   window_size = 7,
+   smoothness = c(5, 1, 1, 2, 4, 1)
+ )
```

To see the impact of small  $\sigma^2$ , we compute the labeled map.

```
R> map_smooth_low <- bayes_label(smooth_low)
R> bayes_map(map_smooth_low)
```

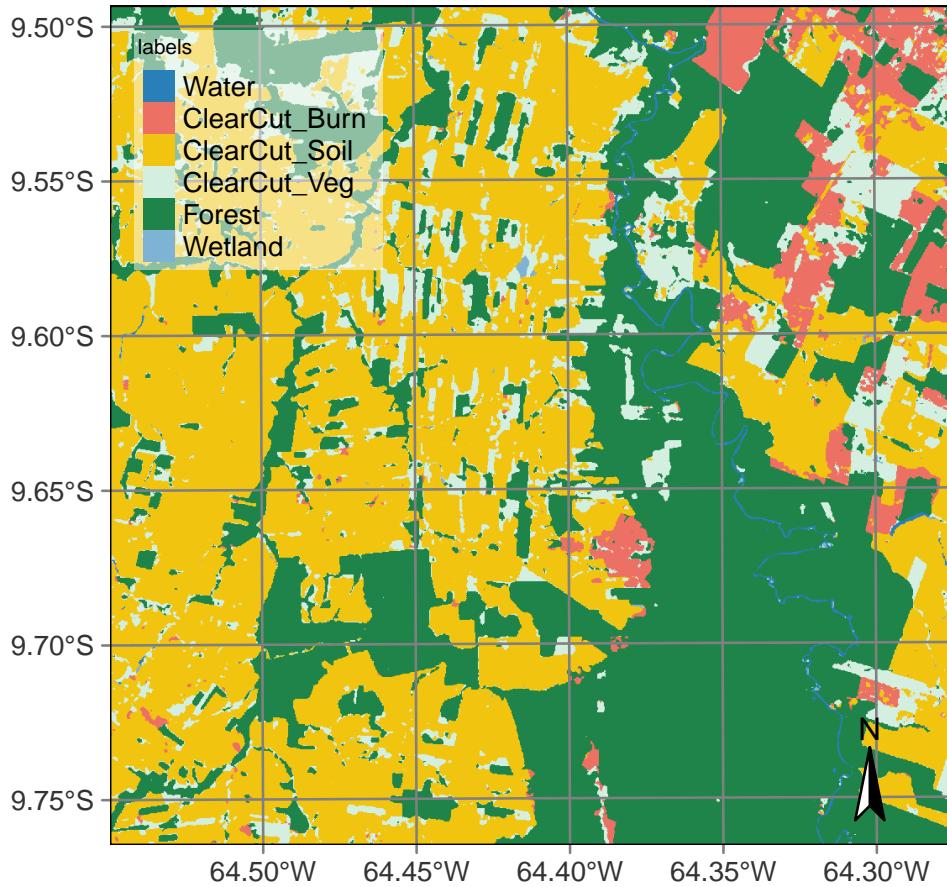


Figure 7: Labeled map with low smoothing parameters.

A visual comparison between the two smoothed maps shows that there is an increase in the area of the ClearCut\_Veg class. Such observation is confirmed by comparing the class areas of the non-smoothed map with the two types of smoothed maps, as shown below.

```
R> sum1 <- bayes_summary(map_no_smooth)
R> colnames(sum1) <- c("class", "area_k2_no_smooth")
R> sum2 <- bayes_summary(map_smooth_high)
R> colnames(sum2) <- c("class", "area_k2_smooth_high")
R> sum3 <- bayes_summary(map_smooth_low)
R> colnames(sum3) <- c("class", "area_k2_smooth_low")
R> dplyr::inner_join(sum1, sum2, by = "class") |>
+   dplyr::inner_join(sum3, by = "class")

# A tibble: 6 x 4
  class      area_k2_no_smooth area_k2_smooth_high area_k2_smooth_low
  <chr>            <dbl>              <dbl>              <dbl>
1 Water             6.26               4.15              4.72
2 ClearCut_Burn    52.6                43.2              44.6 
3 ClearCut_Soil    391.                417.              404. 
```

4 ClearCut_Veg	112.	84.2	97.2
5 Forest	334.	351.	348.
6 Wetland	4.04	0.8	1.43

### 3.4. Relevance of expert knowledge in Bayesian inference

In the smoothed map with higher  $\sigma^2$  values (Figure 6), the most frequent classes (ClearCut\_Soil and Forest) increased their areas at the expense of the others. As shown in Figure 2, these classes occur in more compact patches than the others. In the second smoothed map (Figure 7), there is an increase in the area occupied by the ClearCutVeg class. This increase is due to the nature of this class, which represents a transition between a natural tropical area and one where all trees have been removed. Depending on the aims and practices of those responsible for deforestation, these areas may either have their tree cover removed completely. There are cases, however, where these places are abandoned and turn into secondary vegetation areas [Uhl et al. \(1988\); Wang et al. \(2020\)](#).

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.

Based on the experience of the authors with different experts on land use classification, there are two main approaches for setting the  $\sigma_k^2$  parameter:

1. Increase the neighborhood influence compared with the probability values for each pixel, setting high values (20 or above) to  $\sigma_k^2$  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  $\sigma_k^2$ . 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  $\sigma^2$  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  $\sigma^2$  to be as low as possible. Therefore, the choice of  $\sigma^2$  depends on the effect intended by the expert in the final classified map.

## 4. Comparison with other methods

## Conclusion

- 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.
- Cressie N, Wikle C (2011). *Statistics for Spatio-Temporal Data* | Wiley. Wiley, Hoboken, NJ.
- 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.
- Gong P, Howarth PJ (1989). “Performance Analyses of Probabilistic Relaxation Methods for Land-Cover Classification.” *Remote Sensing of Environment*, **30**(1), 33–42. ISSN 0034-4257. doi: [10.1016/0034-4257\(89\)90045-X](https://doi.org/10.1016/0034-4257(89)90045-X).
- 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.
- 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).
- Mountrakis G, Im J, Ogole C (2011). “Support Vector Machines in Remote Sensing: A Review.” *ISPRS Journal of Photogrammetry and Remote Sensing*, **66**(3), 247–259.
- 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.
- Uhl C, Buschbacher R, Serrao EAS (1988). “Abandoned Pastures in Eastern Amazonia. I. Patterns of Plant Succession.” *Journal of Ecology*, **76**(3), 663–681. ISSN 0022-0477. doi: [10.2307/2260566](https://doi.org/10.2307/2260566).
- Wang Y, Ziv G, Adami M, de Almeida CA, Antunes JFG, Coutinho AC, Esquerdo JCDM, Gomes AR, Galbraith D (2020). “Upturn in Secondary Forest Clearing Buffers Primary Forest Loss in the Brazilian Amazon.” *Nature Sustainability*, **3**(4), 290–295. ISSN 2398-9629. doi: [10.1038/s41893-019-0470-4](https://doi.org/10.1038/s41893-019-0470-4).

### 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](mailto:gilberto.camara@inpe.br)