



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

Remote sensing images are crucial to environmental management. Its uses include measuring changes in land cover, agriculture, and natural habitats. Images identify and track wildfires, enabling timely response and effective response. Satellite measures are the only way to provide a systematic cover of large and distant areas; an example is detection of tropical deforestation, which is crucial in the fight against climate change. By providing vital information to policymakers and conservationists, remote sensing data helps protect the world's environment.

One of the most important tasks in remote sensing image processing is image classification, a task which partitions the pixels in an image into different classes based on their spectral characteristics, spatial patterns, or other relevant features. Machine learning methods such as random forests ([Belgiu and Dragut 2016](#)) and deep learning ([Ma et al. 2019](#)) are commonly used for this job. These techniques typically rely on training samples that are derived from single and isolated pixels, hand-picked by users to represent the desired output classes. The result of machine learning classification is usually a set of matrices with the same size as the original image. Each matrix contains the likelihoods for that a pixel belongs to one of the classes.

Post-processing techniques have become crucial to refine the labels of a classified image ([Huang et al. 2014](#)). There are a number of reasons why the post-processing is required. Classification algorithms may introduce noise or produce misclassifications due to the complexity of remote sensing data. At any resolution level, an image pixel only summarizes what is real. Pixels with mixed landscapes are common and this can misguide a classification technique. The pixel resolution is the size of the ground area that each pixel represents in the image. This ground area may not be homogeneously composed of a single class, especially on transition zones between different landscapes. The features associated with a given pixel are an amalgamation or fusion of the diverse elements on these mixed pixels and they may be hard to classify.

Another reason for post-processing is to refine the boundary between classes or to filter out outliers and misclassified pixels. The classification algorithms may produce noisy or fragmented class assignments, especially when dealing with high-resolution imagery. The features used to classify the pixels, such as their spectral characteristics, have large variability and different classes show great overlap. The consequence is that the classifiers often produce results with outliers or misclassified pixels. By mitigating these errors and minimizing noise, post-processing improves the quality of the initial classification results, bringing a significant gain in the overall accuracy and interpretability of the final output ([Schindler 2012](#)).

Both reasons for post-processing are clearly seen in Figure 1, which shows a fragment of a Sentinel-2 image from the Amazon 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). However, we see the border pixels between different land classes with less stark colors representing their fuzzy class assignment.

Overall, post-processing of machine learning output in remote sensing image classification is essential to enhance the accuracy, spatial coherence, and interpretability of the classification results, ensuring that the final maps provide reliable information for decision-making and subsequent analysis. 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, semi-global, bilateral and edge-aware filtering ([Schindler 2012](#)), modal filters ([Ghimire et al. 2010](#)), and co-occurrence matrices ([Huang et al. 2014](#)).

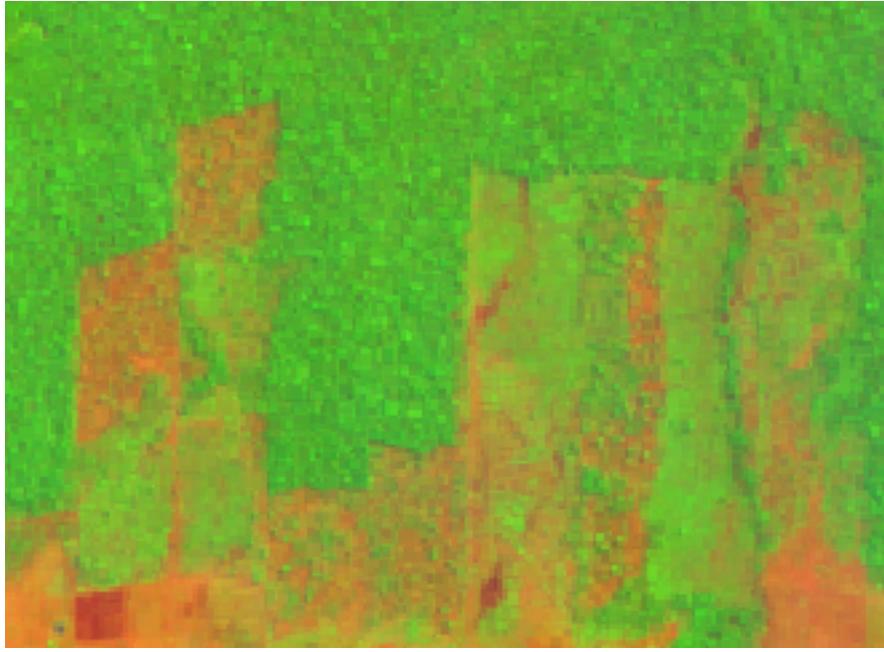


Figure 1: Sentinel-2 image of deforestation in the Amazon forest (detail).

Despite their usefulness, these methods have shortcomings. Gaussian and semi-global smoothing assumes a single value of local variance per class for all neighborhoods. These methods also consider that pixel likelihoods change slowly within a neighborhood, and that the correlation depends only on how far two pixels are from each other (Schindler 2012). Bilateral smoothing is an *ad-hoc* method that introduces a global parameter that tries to preserve the shape of boundaries which are lost by Gaussian filters. The co-occurrence matrices proposed by Huang *et al.* (2014) work on categorical images and do not consider the likelihood values for each pixel. Also, these methods assume the spatial distribution of the class likelihoods is isotropic. When using the likelihood of a pixel and its neighbors to compute properties such as mean, variance and spatial autocorrelation, all spatial directions are considered equivalent. As we will argue in this paper, such assumption does not hold in general in the case of border pixels.

In this paper, we introduced a new type of post-processing algorithm founded on an Empirical Bayes approach and catered to the specific properties of land classification. The algorithm has been implemented in an R package, **bayesEO**. 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. The **bayesEO** package can be used jointly 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 the 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) include 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 are aware, there is no

current R package that 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.

1.1. Methods

In land classification, we deal with categorical data, with each category corresponding to a different land type (e.g., forest, grassland, water body). 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

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

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. The left side of Figure 2 shows such assignment produced by a random forest algorithm and explained in detail in section *Software and examples*. However, this assignment can be problematic when the highest probability does not significantly surpass the others, resulting in an assignment with low confidence. This happens more frequently in pixels located at the transition zones between classes leading to noisy or fragmented class assignments.

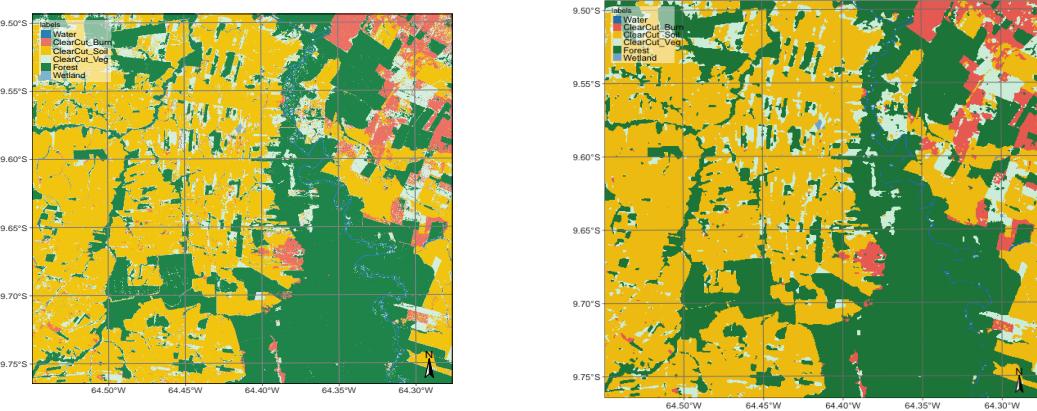


Figure 2: Labelled maps without smoothing (left) and after smoothing (right)

Usually, the spatial \mathbf{s}_i 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 .

The main idea behind our post-processing method is that a pixel classification should take into account its neighborhood pixels. The right side of Figure 2 shows the end result of applying our method on the random forest classification method shown in the left. We want to smooth the output and generate more coherent class representations, reducing the effects of small-scale variations or misclassifications. As the training dataset has no spatial context for the single pixels in T , most remote sensing classification algorithms produce pixel-level classifications without considering the spatial coherence of neighboring pixels. This spatial coherence is introduced as a post-processing step.

1.2. A Bayesian approach

We initially consider a Bayesian approach for this task. 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

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

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 assumes a Gaussian distribution conditionally on $\mu_{i,k}$: $x_{i,k} = \log(p_{i,k}/(1-p_{i,k})) \sim N(\mu_{i,k}, \sigma_k^2)$. 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

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

The posterior mean, expressed as a weighted average between the pixel value $x_{i,k}$ and the prior mean $m_{i,k}$, plays a crucial role. When the prior variance $s_{i,k}^2$ is high, the algorithm assigns more weight to the pixel value $x_{i,k}$. Conversely, as the likelihood variance σ_k^2 increases, the method assigns more weight to the prior mean $m_{i,k}$.

There are three aspects that must be taken into account to proceed. First, there is usually no prior information to specify $m_{i,k}$ and $s_{i,k}^2$. Because of that, we will adopt an Empirical Bayes (EB) approach to obtain estimates of these prior parameters. Second, this method is not considering the pixel neighborhood. To deal with this, we will apply the EB locally in a neighborhood of each pixel.

The third aspect is the most relevant one. A plain neighborhood of each pixel, based uniquely on the distance between locations, did not produce reasonable results. We must consider non-isotropic neighborhoods $\mathcal{N}_{i,k}$ specific for each class k and pixel i . The reason is that we want to give more weight to those neighborhood pixels with assignment classes with large probabilities as those are the ones estimated more reliably by the classification algorithm. 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

$$\hat{m}_{i,k} = \frac{1}{1-\alpha} \int_{\alpha}^{\infty} \mathbb{F}_{i,k}^{-1}(s) ds, \quad (3)$$

the average of the largest $(1-\alpha)$ -th fraction order statistics of the $p_{i,k}$ logit transformed observations. Likewise, based on the 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 (2).

1.3. 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 pixel 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 neighborhood.

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 probability 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 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.

1.4. 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 7×7 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 7×7 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.

2. Software and examples

2.1. 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 3 shows the plot of all layers 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)
```

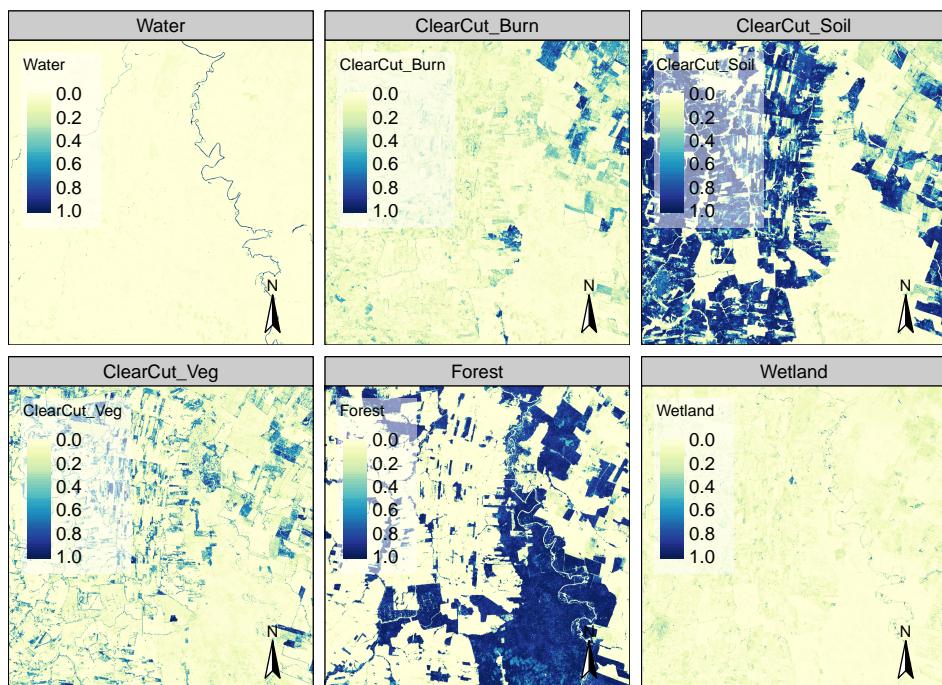


Figure 3: Class probabilities produced by random forest algorithm.

Figure 4 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)
```

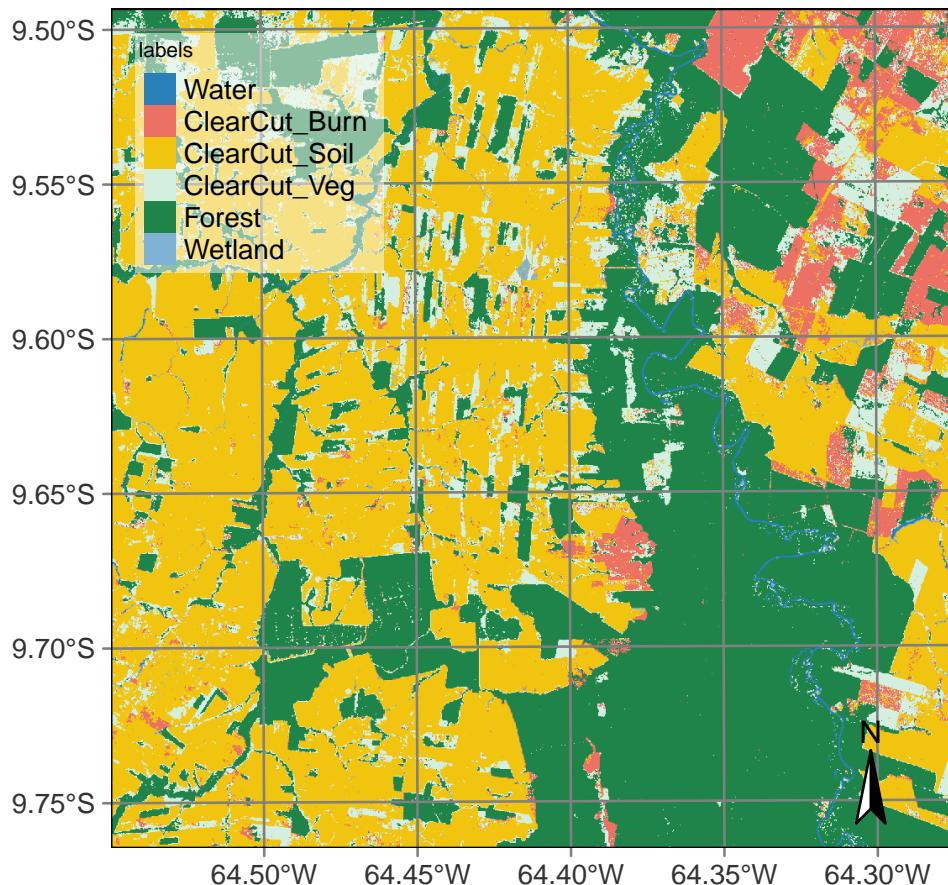


Figure 4: Labelled map without smoothing.

2.2. 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 7×7 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. : 0.001214	Min. : 0.00000	Min. : 0.00000
1st Qu.: 0.1612	1st Qu.: 0.068792	1st Qu.: 0.09474	1st Qu.: 0.09772
Median : 2.0534	Median : 0.159645	Median : 0.23852	Median : 0.24155
Mean : 2.4927	Mean : 0.440458	Mean : 0.84096	Mean : 0.68647
3rd Qu.: 4.6397	3rd Qu.: 0.350913	3rd Qu.: 0.64510	3rd Qu.: 0.71808
Max. :22.2649	Max. :11.051078	Max. :10.59077	Max. :19.33537
Forest	Wetland		
Min. : 0.0000	Min. : 0.00000		
1st Qu.: 0.1026	1st Qu.: 0.07217		
Median : 0.4152	Median : 0.17997		
Mean : 1.5687	Mean : 0.71777		
3rd Qu.: 2.2834	3rd Qu.: 0.42095		
Max. :25.7454	Max. :10.66211		

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 5 shows the values of local logit variances for classes `ClearCut_Soil` and `Forest`, considering only the 4th 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"))
```

Comparing the logit variance maps of Figure 5 with the probability maps of Figure 3 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

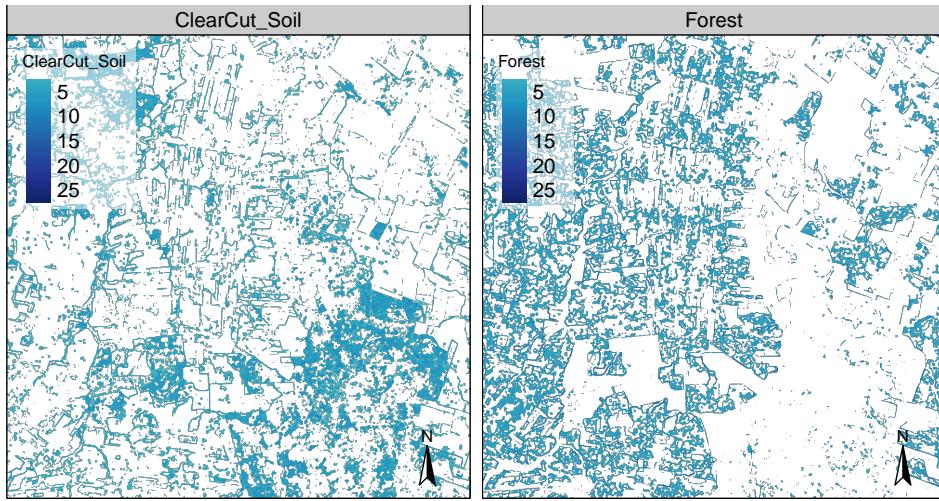


Figure 5: Logit variance map showing values above the 3rd quartile.

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 6, which shows the histograms of local variances per class. The values shown correspond to the 4th 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 4th quartile of class ClearCut_Soil have low values.

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

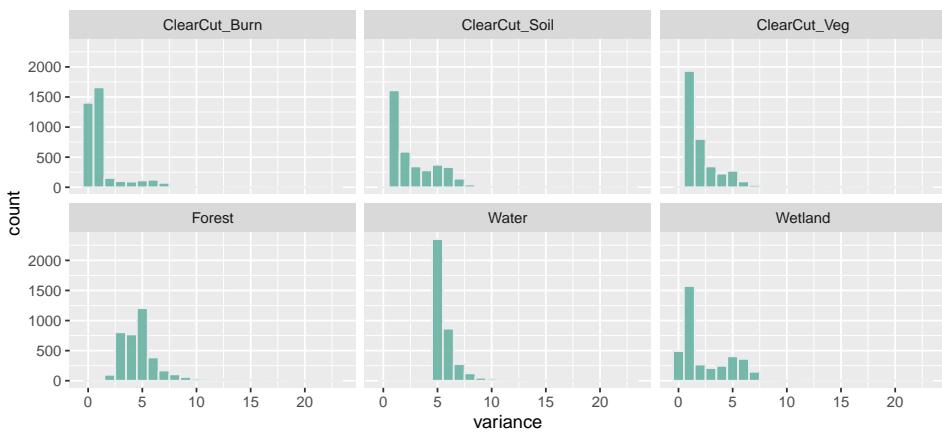


Figure 6: Histogram of top quartile of variances of class logits.

2.3. Applying Bayesian smoothing to remove outliers

As discussed above, the effect of the Bayesian estimator depends on the values of the a priori variance σ_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 σ^2 values we consider two cases: (a) setting σ^2 to a high value close to the maximum value of local logit variance; (b) setting σ^2 to a lower value close to the minimum value of the 4th 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 σ^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 7 shows that the outliers and isolated pixels have been removed.

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

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 4th 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 σ^2 , we compute the labeled map.

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

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)
```

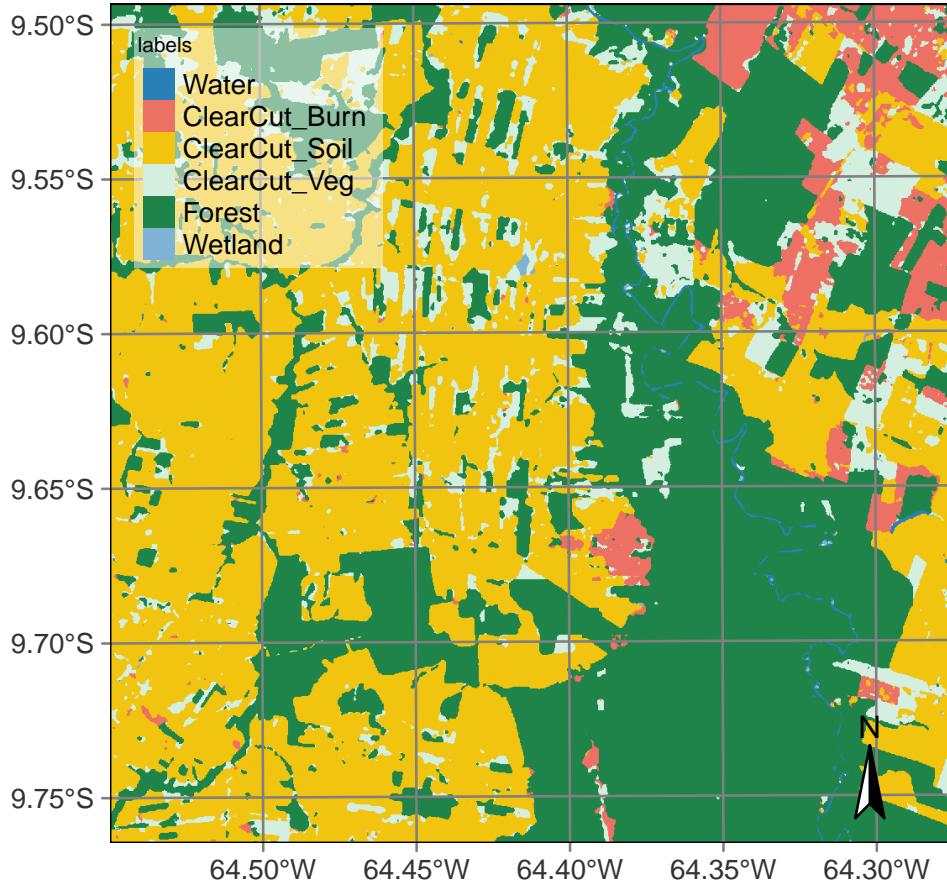


Figure 7: Labeled map with smoothing with high smoothness values.

```
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
```

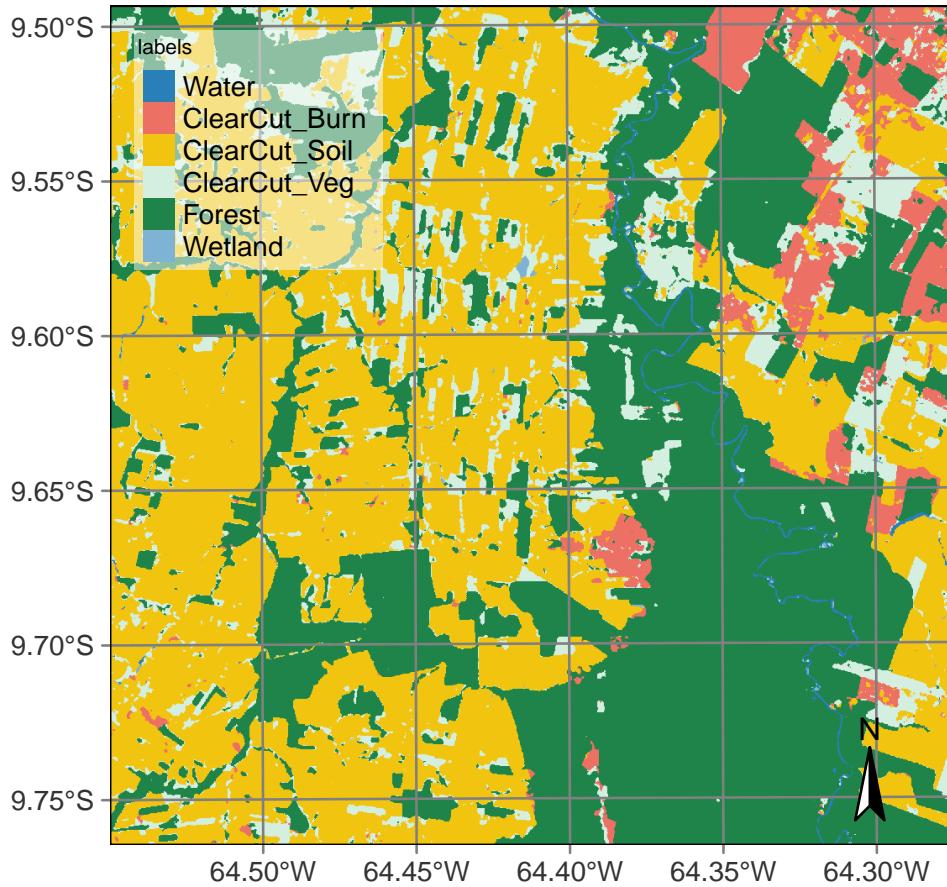


Figure 8: Labeled map with low smoothing parameters.

2.4. Relevance of expert knowledge in Bayesian inference

In the smoothed map with higher σ^2 values (Figure 7), the most frequent classes (ClearCut_Soil and Forest) increased their areas at the expense of the others. As shown in Figure 3, these classes occur in more compact patches than the others. In the second smoothed map (Figure 8), 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 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.

3. 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.
- 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](https://doi.org/10.1201/9780429459016).
- 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.
- 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).

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).

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