Journal of Statistical Software

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

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

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 the popular for remote sensing image classification. Images resulting from these classifiers frequently have outliers or misclassified pixels. For this reason, image post-processing techniques are widely used to refine the labelling in a classified image in order to enhance its classification accuracy.

The `bayesEO` package provides a new method for Bayesian post-processing of images produced by machine learning algorithms. The input to the package is an image containing the probabilities of that pixel belonging to each of the classes. The package provides efficient methods for removing outliers and improving class labelling.

Probability-based smoothing methods include Gaussian and edge-aware filtering (Schindler 2012), modal filters (Ghimire *et al.* 2010), and probabilistic relaxation (Gong and Howarth 1989). Huang *et al.* (2014) propose a relearning method based on co-occurrence matrices; these matrices represent the joint distribution of class labels in the neighborhood of each pixel. In the current work, we introduce a Bayesian smoothing method, which provides the means to incorporate prior knowledge in data analysis.

Bayesian inference can be thought of as way of coherently update our uncertainty in the light of new evidence. It can consider expert knowledge on the derivation of probabilities. As stated by Spiegelhalter and Rice (2009): "In the Bayesian paradigm, degrees of belief in states of nature are specified. Bayesian statistical methods start with existing 'prior' beliefs, and update these using data to give 'posterior' beliefs, which may be used as the basis for inferential decisions".

Bayesian smoothing methods have become an established technique for image restoration and reconstruction (Hanson 1993). More recently, Wu *et al.* (2017) proposed a Bayesian approach for soft fusion applied to change detection. However, to the best of our knowledge, this work is the first proposal to use Bayesian smoothing for post-processing of machine learning pixel based classification.

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

The **sits** package uses satellite image time series for land classification, using
a time-first, space-later approach Simoes *et al.* (2021). In the data preparation part, collections of big Earth observation images are organized as data cubes. Each spatial location of a data cube is associated with a time series. Locations with known labels train a machine learning classifier, which classifies all time series of a data cube. The package has tools for analysis, visualization, and classification of satellite image time series.

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 $p_{i,k} >= 0$ be the probability of pixel i belonging to class k, $k = 1, \dots, K$, where K is the number of classes, then

$$\sum_{k=1}^{K} p_{i,k} = 1. \tag{1}$$

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

$$x_{i,k} = \ln\left(\frac{p_{i,k}}{1-p_{i,k}}\right). \tag{2}$$

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

After transforming probability values into logits, the problem can be expressed in a Bayesian context. Suppose that $x \in X$ has a distribution $\pi(x|\theta)$ with an unknown $\theta \in \Theta$ parameter. Consider two random variables for each pixel i and class k: (a) $x_{i,k}$, the observed class logits; (b) $\mu_{i,k}$, the inferred logit values based on the observations. From the output of the machine learning classification is $x_{i,k}$, the aim is to estimate the actual values $\mu_{i,k}|x_{i,k}$. The Bayesian inference procedure can be expressed as

$$\pi(\mu|x) \propto \pi(x|\mu)\pi(\mu). \tag{3}$$

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 underlying values $\mu_{i,k}$; and (b) $\pi(\mu)$, which is our guess on the actual data distribution, known as the prior. For simplicity, independence between the different classes k is assumed. Therefore, 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 $N(\mu_{i,k},\sigma^2_k)$, with parameters $\mu_{i,k}$ and σ^2_k. The variance σ^2_k will be estimated based on user expertise and taken as a hyperparameter to control the smoothness of the resulting estimate. Therefore

$$x_{i,k}|\mu_{i,k} \sim N(\mu_{i,k},\sigma^2_k) \tag{4}$$

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

$$\mu_{i,k} \sim N(m_{i,k}, s^2_{i,k}). \tag{5}$$

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

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

and the variance by

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

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

$$E[\mu_{i,k}|x_{i,k}] = \frac{m_{i,t} \times \sigma^2_k + x_{i,k} \times s^2_{i,k}}{\sigma^2_k + s^2_{i,k}}, \tag{8}$$

which can be expressed as a weighted mean

$$E[\mu_{i,k}|x_{i,k}] = \left[\frac{s^2_{i,k}}{\sigma^2_k + s^2_{i,k}}\right] \times x_{i,k} + \left[\frac{\sigma^2_k}{\sigma^2_k + s^2_{i,k}}\right] \times m_{i,k}, \tag{9}$$

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.
- $s^2_{i,k}$ is the variance of logit values for pixels of class k in the neighborhood of pixel i.
- σ^2_k 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 $s^2_{i,k}$ for the neighbors is too high, the algorithm gives more weight to the pixel value $x_{i,k}$. When the variance of the likelihood σ^2_k increases, the method gives more weight to the neighborhood mean $m_{i,k}$.

2.3. 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 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 σ_k^2, representing prior belief in the variability of the probability values for each class. If we take both σ_A^2 for class A and σ_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 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.

2.4. Defining the neighbourhood

The underlying principle of Bayesian smoothing is that regions with similar characteristics typically have a dominant class, characterized by higher average probabilities and lower variance compared to other classes. In such areas, a pixel belonging to a different class is expected to exhibit lower average probabilities and higher local variance. Consequently, post-processing should alter the class of this pixel to match the dominant class.

Classification challenges arise when pixels are situated along the boundaries between areas containing different classes, as they possess signatures of two classes. To ensure the reliability of local class statistics, only pixels with a high likelihood of belonging to the class should be included. In the context of spatial neighborhoods centered on border pixels, only a portion of the pixels in the window share the same class as the central pixel, while the others belong to a different class. To account for border pixels, we use a specialized definition of a neighborhood to estimate the prior class distribution.

To approximate the prior distribution using spatial neighborhoods, the algorithm uses pixels with high probability of belonging to the class. The rationale is that pixels with low probabilities are unlikely to be part of the same class distribution as those with high probability values. This procedure ensures a more reliable estimation of the prior.

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 forests 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 ro 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 1 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)
```

The non-smoothed labeled map shows the need for post-processing. This map is obtained by taking the class of higher probability to each pixel, without considering the spatial context. This is done by `bayes_label()` whose main parameters are: (a) `x`, a SpatRaster object containing the probabilities of each class for all pixels; (b) `labels`, the labels associated to the classes.

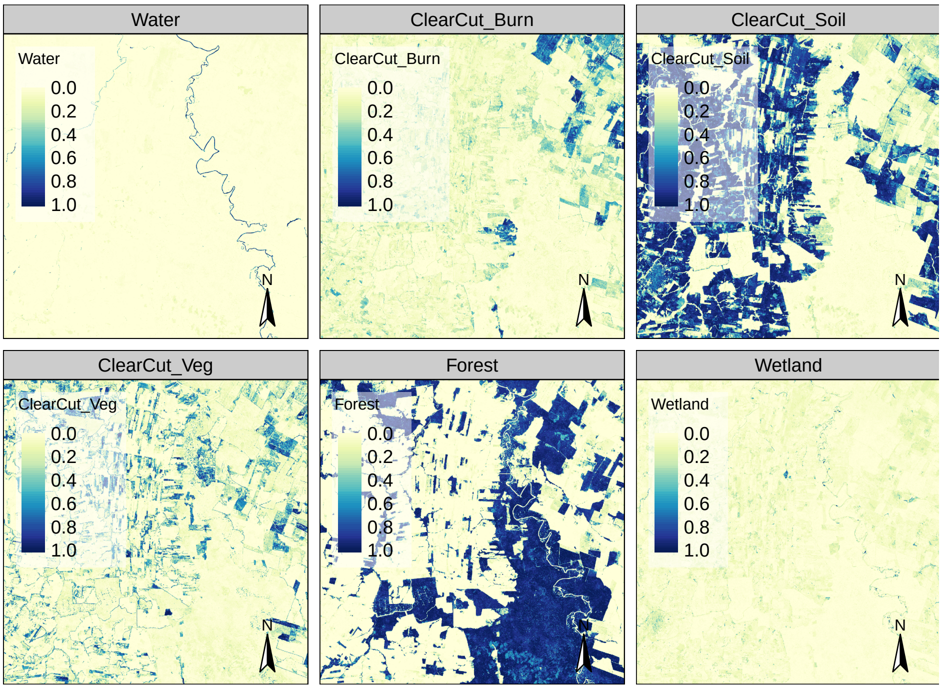

Figure 1: Class probabilities produced by random forest algorithm.

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

Figure 2 shows the resulting map, which contains a significant number of outliers and misclassified pixels.

```
R> bayes_map(map_no_smooth,
+           tmap_legend_title_size = 0.7,
+           tmap_legend_text_size = 0.7)
```

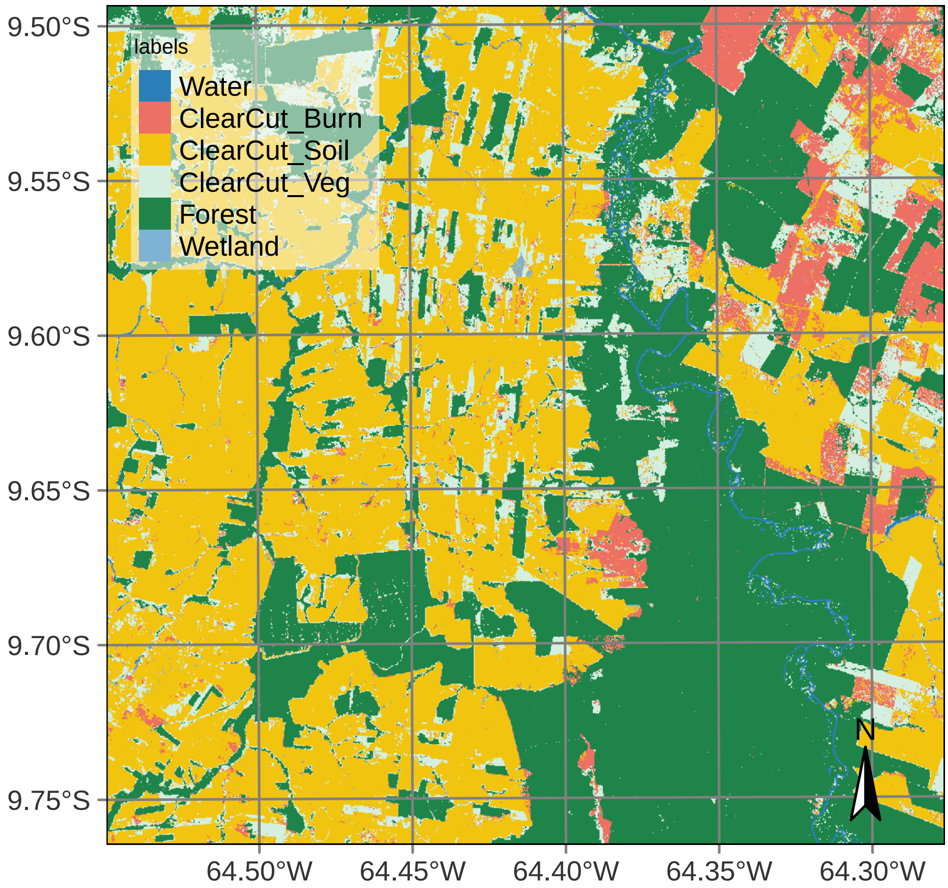

Figure 2: Labelled map without smoothing.

3.2. Estimating the local logit variances

The local logit variances correspond to the $s^2_{i,k}$ parameter in the Bayesian inference and are estimated by `sits_variance()`. Its main parameters are: (a) `cube`, a probability cube; (b) `window_size`, dimension of the local neighborhood; (c) `neigh_fraction`, the percentage of pixels in the neighborhood used to calculate the variance; (d) `output_dir`, directory where results will be stored. 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 = 0.5)
R> bayes_summary(var_image)
```

```
 Water              ClearCut_Burn       ClearCut_Soil       ClearCut_Veg
 Min.   : 0.0000    Min.   : 0.00296    Min.   : 0.00000    Min.   : 0.00000
 1st Qu.: 0.1652    1st Qu.: 0.06693    1st Qu.: 0.09558    1st Qu.: 0.09659
 Median : 2.2109    Median : 0.15768    Median : 0.23407    Median : 0.24058
```

```
 Mean   : 2.4892   Mean   : 0.41758   Mean   : 0.84083   Mean   : 0.69320
 3rd Qu.: 4.6471   3rd Qu.: 0.34780   3rd Qu.: 0.63066   3rd Qu.: 0.71301
 Max.   :24.2813   Max.   :11.10299   Max.   :13.82341   Max.   :16.61255
 Forest            Wetland
 Min.   : 0.0000   Min.   : 0.00000
 1st Qu.: 0.1042   1st Qu.: 0.07132
 Median : 0.4069   Median : 0.17736
 Mean   : 1.5824   Mean   : 0.69689
 3rd Qu.: 2.3758   3rd Qu.: 0.41690
 Max.   :23.5609   Max.   :10.91042
```

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 3 shows the values of local logit variances for classes `ClearCut_Soil` and `Forest`, considering only the 4^{th} quartile of the distribution. Only the top 25% of the values for each class are shown, emphasizing areas of high local variability.

```
R> bayes_plot(var_image, quantile = 0.75, labels = c("ClearCut_Soil", "Forest"))
```

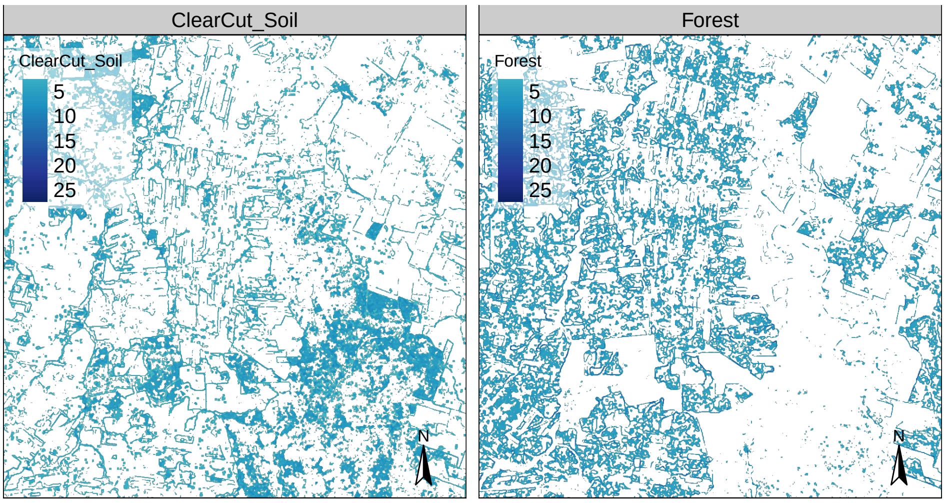

Figure 3: Logit variance map showing values above the 3rd quartile.

Comparing the logit variance maps of Figure 3 with the probability maps of Figure 1 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 4, which shows the histograms of local variances per class. The values shown correspond to the 4^{th} 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^{th} quartile of class `ClearCut_Soil` have low values.

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

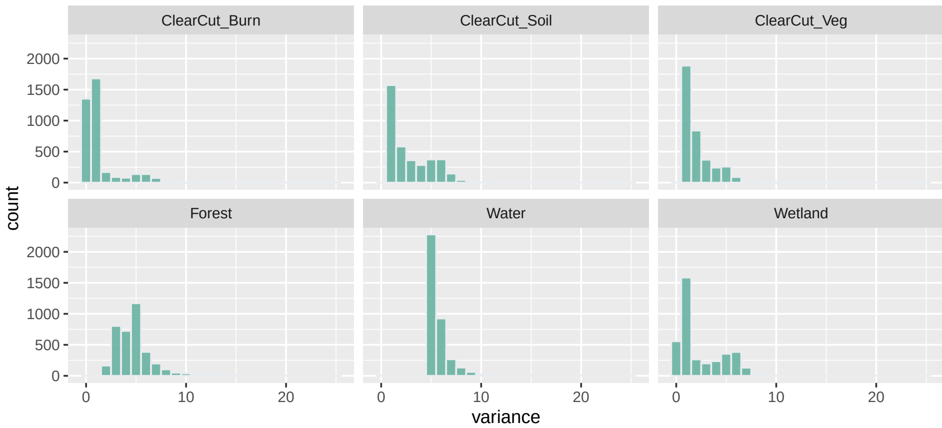

Figure 4: Histogram of top quartile of variances of class logits.

3.3. Applying Bayesian smoothing to remove outliers

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 neighborhood; (c) `neigh_fraction`, percentage of pixels in the neighborhood used to calculate the variance; (d) `smoothness`, prior logit variances for each class.

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 neighborhood 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 4^{th} quartile.

```
R> smooth_high <- bayes_smooth(
+     probs_image,
+     window_size = 7,
+     neigh_fraction = 0.5,
+     smoothness = c(25, 10, 20, 20, 25, 10)
+ )
```

The impact of Bayesian smoothing can be best captured by producing a labelled map using `sits_label_classification()`, taking the smoothed image as its input. Figure 5 shows that the outliers and isolated pixels have been removed.

```
R> map_smooth_high <- bayes_label(smooth_high)
R> bayes_map(map_smooth_high,
+           tmap_legend_title_size = 0.7,
+           tmap_legend_text_size = 0.7)
```

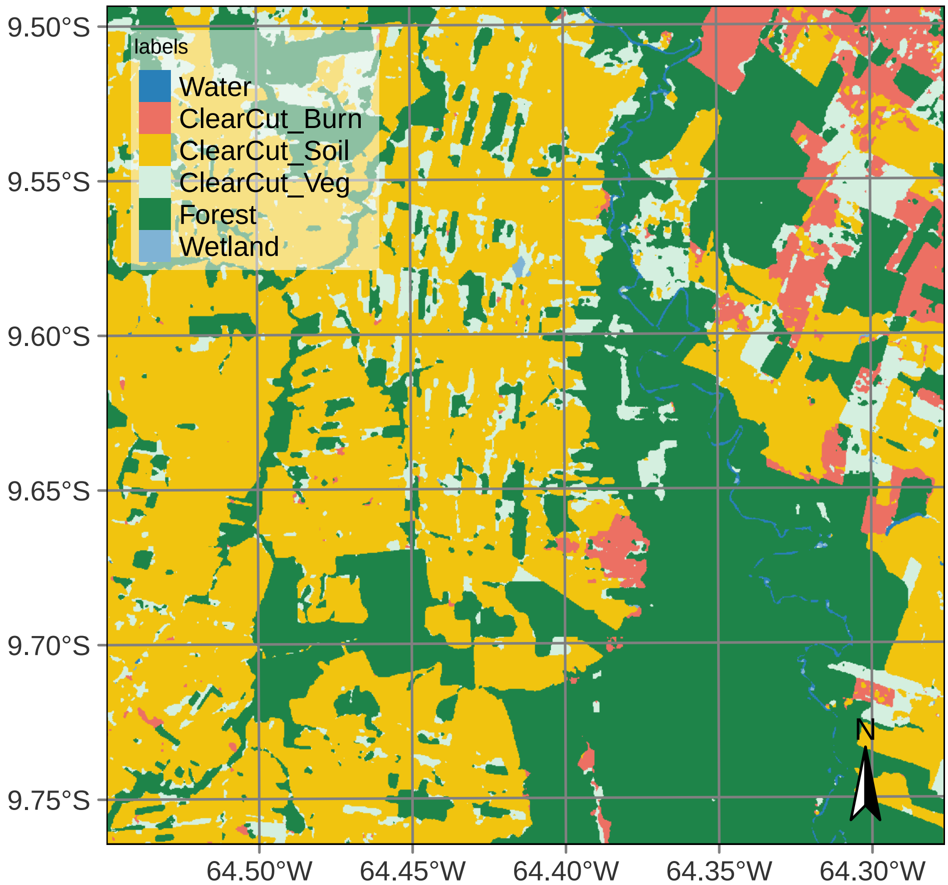

Figure 5: Labeled map with smoothing with high smoothness values.

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^{th} quartile of the local variance for each class, as computed below.

```
R> smooth_low <- bayes_smooth(
+     probs_image,
+     window_size = 7,
+     neigh_fraction = 0.5,
+     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,
+            tmap_legend_title_size = 0.7,
+            tmap_legend_text_size = 0.7)
```

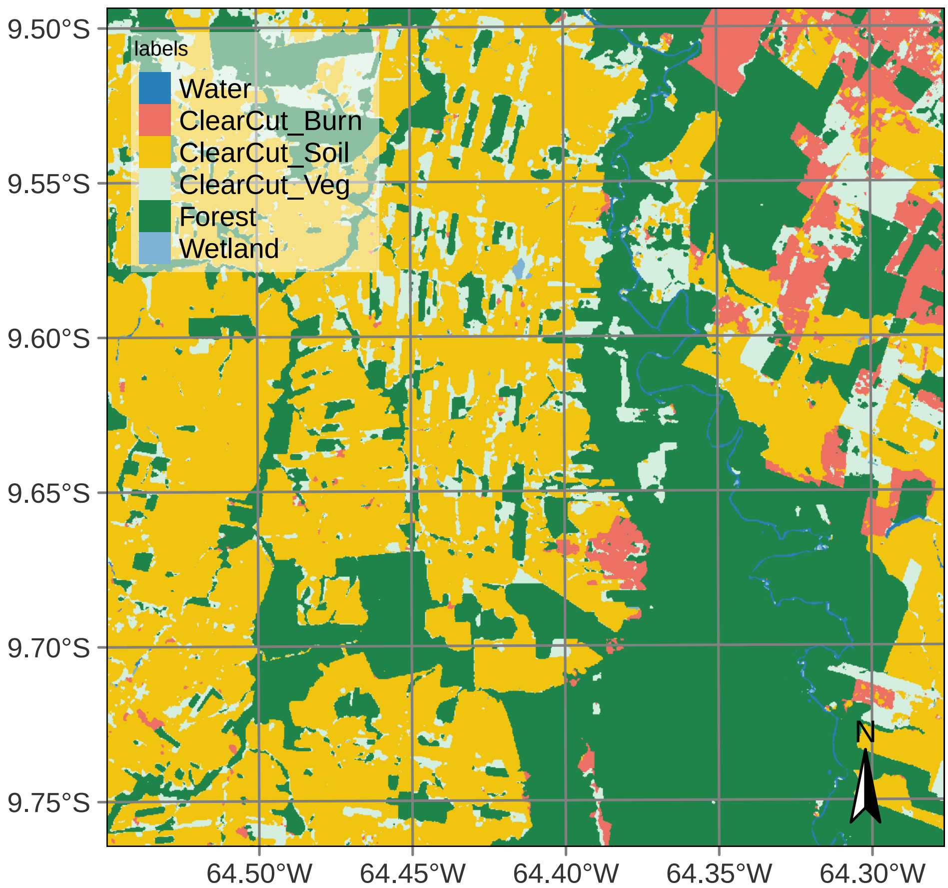

Figure 6: Labeled map with low smoothing parameters.

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 prior variance values, the most frequent classes (`ClearCut_Soil` and `Forest`) increased their areas at the expense of the others. As shown in Figure 1, these classes occur in more compact patches than the others. In the second smoothed map, 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.

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.

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.

Hanson KM (1993). "Introduction to Bayesian Image Analysis." In *Medical Imaging 1993: Image Processing*, volume 1898, pp. 716–731. SPIE. doi:10.1117/12.154577.

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.

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.

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.

Spiegelhalter D, Rice K (2009). "Bayesian Statistics." *Scholarpedia*, **4**(8).

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.

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.

Wu C, Du B, Cui X, Zhang L (2017). "A Post-Classification Change Detection Method Based on Iterative Slow Feature Analysis and Bayesian Soft Fusion." *Remote Sensing of Environment*, **199**, 241–255. ISSN 0034-4257. doi:10.1016/j.rse.2017.07.009.

Affiliation:

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

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