J. Approx. Softw., Vol. 2, No. 1 (2025) 1

Journal of Approximation Software

doi: 10.13135/3103-1935/11095 – ISSN: 3103–1935

Bayesian-STAPLE: a python module for Bayesian label fusion

D. Cazzorla [1] and C. Mencar [1,⋆]

[1] *Dipartimento di Informatica, Università degli Studi di Bari Aldo Moro, Bari, Italy*

Received: 23/09/2024 – Published: 26/03/2025
Communicated by: M. Vianello

Abstract

Label fusion is widely used in fields such as medical imaging, remote sensing, and product rating, among others. In a label-fusion application, usually several assessments of the same item are obtained, often from different algorithms or human experts, to estimate the missing ground truth. However, a quantification of performance uncertainty is important to understand whether more data (e.g., more raters) are needed for a more certain estimation of the ground truth. In this work, we describe a software that implements our Bayesian extension of STAPLE in order to provide an estimation of uncertainty, which could help decision makers in accepting estimations or requiring additional data. The main feature of the developed software is its wider applicability in many contexts where ground truth is not available, but can be estimated by reaching a consensus among different rates. The experimental results on the MSSEG2016 dataset on brain segmentations to identify multiple sclerosis lesions are reported to show the effectiveness of the methods and the use of the software.

Keywords: STAPLE, Label fusion, Bayesian method, Medical image analysis (MSC2020: 62C10, 62H35)

1 Introduction

Label fusion is a technique used in medical imaging and computer vision to combine multiple labeled images or segmentations into a single, more accurate result [15]. It is commonly applied in the context of medical image analysis, such as brain imaging, where different segmentations (e.g., anatomical structures) from various sources or experts are integrated to improve overall precision and reliability of labels [17]. Although medical image analysis is a main field of application, label fusion can have different applications, including remote sensing for Earth

⋆ Corresponding author: corrado.mencar@uniba.it

observation [13], emotion recognition in videos [16], product rating, and web search quality [20]. For the sake of clarity, and without loss of generality, we will focus on medical imaging as a guiding example throughout the paper.

In a label-fusion application, usually several segmentations of the same image are obtained, often from different algorithms or human experts. Then, these segmentations are combined using various statistical or computational techniques to produce a final consensus segmentation. A common method for label fusion is majority voting, where each voxel or pixel is assigned the label that appears most frequently across the different segmentations. In more complex cases, votes from different segmentations can be weighted according to the confidence or accuracy of the source, improving the fusion result. However, the problem of finding an appropriate weighting arises, which leads to advanced label fusion techniques that use statistical models to account for the variability and reliability of the different segmentations. A leading example of advanced label fusion is STAPLE (Simultaneous Truth and Performance Level Estimation) and its variants [1, 2, 8, 12, 18, 19].

Label fusion is extensively used in medical imaging for the segmentation of organs, tumors, and other anatomical structures in modalities such as MRI, CT, and PET. In addition, in neuroimaging, label fusion helps create accurate brain atlases and in the segmentation of brain regions, which is critical for both research and clinical applications. In a more general context, label fusion techniques are also applied in machine learning to generate ground truth labels from multiple annotators [14].

The STAPLE algorithm is a widely-used method for label fusion. It works by iteratively estimating the true segmentation and the performance of each individual segmentation source. Estimation is performed by computing the probabilistic label for each voxel by considering the agreement among the segmentations and the estimated reliability of each rater. One limitation of the original method lies in the estimation of the ground truth, which depends on a *precise* estimation of the performance of the raters. Although this assumption enables a fast estimation of the ground truth, it does not take into account the uncertainty on the performance of the raters. However, a quantification of performance uncertainty is important to understand whether more data (e.g., more raters) are needed for a more certain estimation of the ground truth.

In a previous work, we proposed a fully Bayesian approach to extend STAPLE in order to provide an estimation of uncertainty on the ground truth and the performance of raters [5]. The main goal of the proposed method is to enrich the information provided as a result of the fusion process with uncertainty quantification, which could help decision makers in accepting estimates or requiring additional data. In the same work, we furthermore showed that the results of Bayesian STAPLE are preferable because it does not lose information when compared with a non-Bayesian approach. Another Bayesian extension of STAPLE has been proposed by Audelan et al. [2]. In this case, the authors formulate the problem on a different basis, where the performance of raters is quantified by bias and variance. Our method is closer to the original version of STAPLE, which we believe more interpretable because it is immediate to assess raters in terms of sensitivity and specificity.

A key feature of our method is its wider applicability in many contexts where ground truth is not available, but can be estimated by reaching a consensus among different raters. This enables to cast the proposed method in the realm of ensemble learning, where it could be possible to estimate the performance of each predictive model and assess the reliability of the aggregated decision. For this reason, we developed software that implements our Bayesian extension of STAPLE and made it freely available to the community, so as to promote its application in different areas of Machine Learning and Data Science. The software is available at https://github.com/davideC00/Bayesian-STAPLE with MIT license. Installation requirements

are available on the website.

The paper is organized as follows. In Sect. 2, the problem statement and the Bayesian extension of STAPLE are briefly reported, while Sect. 3 describes the available software, including the code structure. Sect. 4 describes an experimental use case of the software and Sect. 5 draws some final remarks.

2 Method

2.1 Problem statement

A collection of N items, each of unknown class $T_i \in \{0,1\}, i = 1,2,\ldots,N$, are classified by R raters that assign a binary label $D_{ij} \in \{0,1\}$ for $i = 1,2,\ldots,N$ and $j = 1,2,\ldots,R$. (As a guiding example, the items correspond to the voxels of an image that is segmented by raters to detect some anatomical regions of interest; each voxel is labeled as "1" if it belongs to a region, and "0" if it belongs to the background.) Let $\mathbf{D}$ be the matrix $N \times R$ arranging the classifications of all the raters and $\mathbf{T}$ the unknown vector $N \times 1$ representing the ground truth.

It is assumed that each rater has some aleatoric uncertainty in classifying items. This is modeled by introducing raters' *sensitivity* p_j and *specificity* q_j, defined as:

$$p_j = \Pr(D_{ij} = 1 | T_i = 1) \tag{1}$$

and

$$q_j = \Pr(D_{ij} = 0 | T_i = 0) \tag{2}$$

Let $\mathbf{p} = (p_1, p_2, \ldots p_R)$ and $\mathbf{q} = (q_1, q_2, \ldots q_R)$ be the unknown vectors with shape $R \times 1$ representing the sensitivity and specificity of all raters, respectively.

The uncertainty in $(\mathbf{p}, \mathbf{q})$ and the ground truth $\mathbf{T}$ is modeled by considering them as random variables. The raters are assumed to be independent. For the sake of simplicity, we will also assume that the ground truth of each item is independent w.r.t. all other items.

The proposed goal is to estimate the ground truth $\mathbf{T}$ as well as the performance of the raters $\mathbf{p}$ and $\mathbf{q}$, given the observations $\mathbf{D}$.

2.2 Bayesian extension of STAPLE

The method is outlined here, while all technical details can be found in our previous work [5].

2.2.1 Prior distributions

Priors on sensitivities and specificities can be conveniently modeled as Beta functions as they are probabilities:

$$p_j \sim \text{Beta}(\alpha_{p_j}, \beta_{p_j}),\ q_j \sim \text{Beta}(\alpha_{q_j}, \beta_{q_j}) \tag{3}$$

while the ground truth T_i is binary, therefore, it can be modeled as a Bernoulli variable with unknown probability w which, in turn, can be modelled with a Beta distribution:

$$w \sim \text{Beta}\,(\alpha_w, \beta_w) \tag{4}$$

2.2.2 Bayesian model

We make use of Bayesian analysis, through which we define a probabilistic hierarchical model:[1]

$$f(\mathbf{p},\mathbf{q},\mathbf{T},w|\mathbf{D}) = \frac{f(\mathbf{D}|\mathbf{T},\mathbf{p},\mathbf{q},w)f(\mathbf{T},\mathbf{p},\mathbf{q}|w)f(w)}{f(\mathbf{D})} \tag{5}$$

with:

$$f(\mathbf{T},\mathbf{p},\mathbf{q}|w) = f(\mathbf{T}|w)\cdot f(\mathbf{p})\cdot f(\mathbf{q})$$

because we assume pairwise independence between ground truth, sensitivity and specificity of raters. We also assume the independence of the raters and the independence among the items.

The posterior distribution (5) can be estimated numerically via Markov Chain Monte Carlo (MCMC). The structure of the model suggests the adoption of Gibbs sampling for estimating the posterior, which enables the estimation of a joint distribution through its conditionals.

2.2.3 Initialization of hyper-parameters

The Bayesian model requires the specification of the hyperparameters in (3) and (4). Here is the point where the available knowledge can be injected into the model, as in the case of Augmented Data-Dependent Priors (AUDP) or Power Priors [9, 11].

If there is no prior knowledge, it is possible to consider the least informative priors, which require $\alpha_{p_j} = \beta_{p_j} = \alpha_{q_j} = \beta_{q_j} = \alpha_w = \beta_w = 1$, ensuring that all parameters are uniformly distributed between 0 and 1. In practice, we sampled the initial values of p_j and q_j between 0.5 and 1 to avoid the inversion of the rater (for example, a rater with 0 sensitivity and specificity would perfectly label each background voxel as a region of interest, and vice versa). However, this is just a technical trick to start the Gibbs sampler: the subsequent values of p_j and q_j are sampled according to a Beta distribution. Another possible approach to avoid rater inversion is to have α_{p_j} and α_{q_j} greater than 1, but this prior may bias the estimation of the posterior.

3 Software details

3.1 Use case

The main use case of the developed software is the estimation of the ground truth and raters' performance parameters. This use case unfolds in two steps, as described in the following.

3.1.1 Dataset loading

This step is actually external to the software, and the user is responsible for producing a multidimensional array of binary values. Its dimension depends on the data that the raters are labeling. Here are some examples:

- 7 raters labeling an audio divided into 100 intervals → input shape = (100, 7)
- 10 raters labeling a geo-spatial image with 1280 x 1000 → input shape = (1280, 1000, 10)
- 10 raters labeling a video 120x120 of 4300 frames → input shape = (120, 120, 4300, 10)

[1] By f we denote a generic probability measure or probability density function.

In general, the shape is $(\dim_1, \dim_2, ..., \dim_N, \text{raters})$ where the first N dimensions are the dimensions of the data that is labeled and the last dimension refers to the raters. If the raters have labeled the data multiple times (see 3.1.3), the input shape must be

$$(\dim_1, \dim_2, ..., \dim_N, \text{iterations}, \text{raters})$$

3.1.2 Ground truth and performance estimation

Given a dataset D, a model can be instantiated as follows.

```
model = BayesianSTAPLE(D)
```

This operation generates a model object that will contain all the estimation results, which can be obtained by the `sample` method to generate the posterior distribution (5):

```
sample=model.sample(draws, burn_in, chains)
```

The interface of the method is inspired by the sample function of a pymc[2] model. It requires the number of draws to generate, the number of burn-in draws to reach a stable state, and the number of chains for multiple sampling.[3] The output is an array object that can be used in exploratory tools such as ArviZ[4].

After the sampling, the ground truth can be estimated as the expectation of the ground truth's posterior. This can be quickly achieved with the `get_ground_truth` method.

3.1.3 Variants

Prior knowledge on raters (as formalized by the prior distributions described in Sect. 2.2.1) can be specified in the object constructor:

```
model = BayesianSTAPLE(D, alpha_p , beta_p, alpha_q, beta_q)
```

The additional parameters are array-like one-dimensional objects with length equal to the number of raters. When not specified, they are set by default as constant arrays of ones so that the prior distribution coincides with the uniform distribution over $[0, 1]$.

Prior Knowledge on ground truth can be set by specifying the parameter w in (5):

```
model = BayesianSTAPLE(D, w)
```

This can be either a scalar value (indicating a fixed probability for all the items of the ground truth), or an array-like structure (of the same shape of the ground truth) indicating a probability value for any single item.

Since the Bayesian model (5) enables the specification of a probability distribution over w as a Beta distribution, its hyper-parameters can be specified as follows:

```
model = BayesianSTAPLE(D, alpha_w, beta_w)
```

If not specified, their default value is 1 so that the probability distribution is uniform over $[0, 1]$.

[2] https://www.pymc.io

[3] Multiple chains also enable parallel processing, which could be implemented in future versions of the software.

[4] https://python.arviz.org

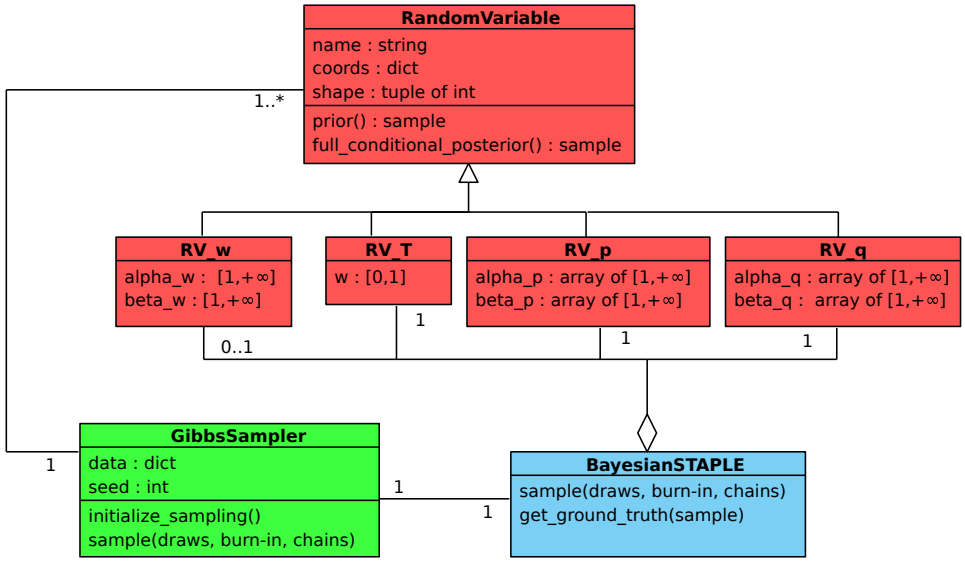


Figure 1: UML diagram of the classes in the module.

Uncertainty reduction by repeated labeling. In this case, the dataset should contain repeated labeling of the same raters (see Sec. 3.1.1) and the class constructor must be informed accordingly:

```
model = BayesianSTAPLE(D, repeated_labeling=True)
```

3.2 Code Structure

The module comprises three main classes, which are described in the following sections. A UML class diagram is depicted in Fig. 1.

3.2.1 Class `BayesianSTAPLE`

This is the class used by the end-user to instantiate a probabilistic model described in Sect. 2.2.2 along with a Gibbs sampler. All the arguments accepted by the constructor can be found in README.md.

Methods in this class are:

- `sample(draws, burn_in, chains)`: Samples from the posterior using the Gibbs sampler.
- `get_ground_truth(sample)`: Receives the generated sample and uses it to estimate the expectation of the ground truth posterior.

3.2.2 class `GibbsSampler`

This class implements Gibbs Sampling as described in [10]. Given the random variables $(\theta_1, \theta_2, ..., \theta_H)$ of the probabilistic model, the Gibbs sampler starts with an initial value for the variables $(\theta_1^{(0)}, \theta_2^{(0)}, ..., \theta_H^{(0)})$ and samples the next values as follows:

$$
\begin{aligned}
&1.\ \theta_1^{(k+1)} \sim f_1(\theta_1|\theta_2^{(k)},\dots,\theta_H^{(k)}) \\
&2.\ \theta_2^{(k+1)} \sim f_2(\theta_2|\theta_1^{(k+1)},\theta_3^{(k)}\dots,\theta_H^{(k)}) \\
&\quad\vdots \\
&H.\ \theta_H^{(k+1)} \sim f_H(\theta_H|\theta_1^{(k+1)},\dots,\theta_{H-1}^{(k+1)})
\end{aligned}
$$

where $f_i(\theta_i|\theta_1,...,\theta_{i-1},\theta_{i+1}...,\theta_n)$ are the full conditional posteriors.

Methods in this class are:

- `initialize_sampling()`: Initialize the value of the random variables.
- `sample(draws, burn-in, chains)`: Sample from the posterior distribution.

3.2.3 Class `RandomVariable`

This abstract class represents a generic random variable. The random variables described in Sect. 2.2.2, are implemented as specialized subclasses: `RV_w`, `RV_T`, `RV_q` and `RV_p`. Methods in this class are:

- `prior()`: Sample of the prior distribution.
- `sample(draws, burn-in, chains)`: Sample of the full conditional distribution.

In order to be used by the Gibbs sampler, the subclasses must override the methods `prior` and `full_conditional_posterior`.

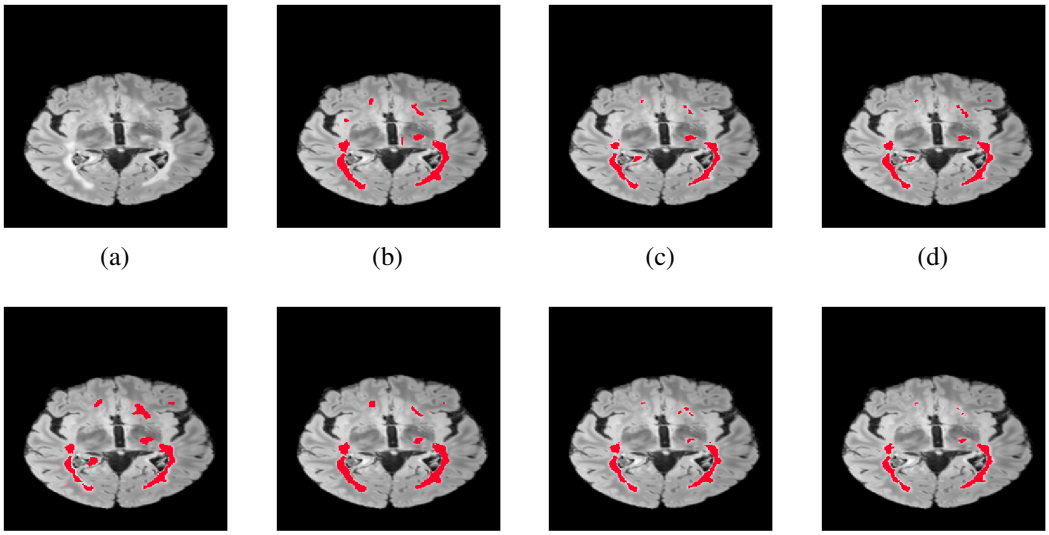
3.3 Implementation details

The module is implemented in Python 3.10 using the JAX library [3]. The use of JAX enables its execution on GPUs and TPUs, resulting in a considerable increase in speed for large inputs. If a GPU or TPU is available, JAX will automatically use it to run the code.

4 Experimental results

The dataset used for this section is the MSSEG 2016 dataset [6, 7]. This dataset contains segmentations of seven junior neurologists who were asked to identify multiple sclerosis lesions on 3D FLAIR images.[5] The segmentations are binary and 256x256x256 wide. As the raters' performance can change for different slices, Bayesian STAPLE was executed separately on each slice. The experiments show how to execute it on slice number 150 (Fig. 2).

[5]Fluid-attenuated inversion recovery (FLAIR) is a special inversion recovery sequence with long inversion time (TI) to remove the effects of fluid on the image. See also https://radiopaedia.org/articles/fluid-attenuated-inversion-recovery.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 2: (a) Slice number 150 of the FLAIR image. (b)-(h) Raters' segmentations. The presence of multiple sclerosis lesion is denoted in red.

4.1 Dataset Loading

The raters' segmentations were stored in NIfTI files and loaded into memory using the `glob` and the `NiBabel` libraries [4].

```
import nibabel as nib
import glob
import numpy as np

raters_data = []
files = sorted(glob.glob(folder_path + 'Rater*'))
for file_name_rater in files:
  rater = nib.load(file_name_rater)
  raters_data.append(np.expand_dims(rater.get_fdata(), axis=-1))
dataset = np.stack( raters_data, axis=-1).astype(bool)
dataset = np.squeeze(dataset)

slice_num=150
D = dataset[:,:,slice_num]
D.shape # (256, 256, 7)
```

4.2 Ground truth and performance estimation

The model was instanced by:

```
model = BayesianSTAPLE(D)
```

The posterior distributions were approximated by:

```
sample = model.sample(1000, burn_in=100, chains=5)
```

To check if there were problems in the convergence, the sampling traces were plot with:

```
import arviz as az
az.plot_trace(sample, var_names=["p","q"], compact=True)
```

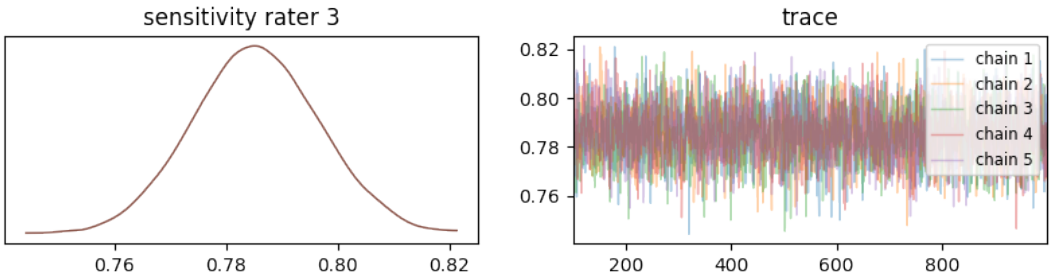


Figure 3: Sensitivity posterior for rater number 3 and its sampling trace.

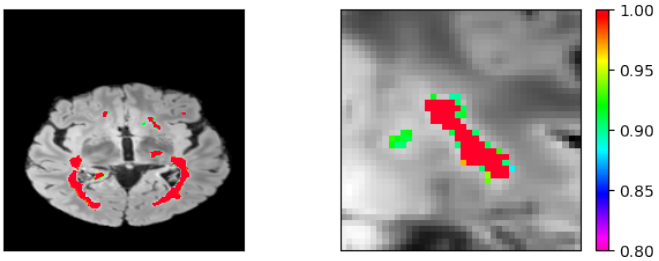


Figure 4: Probability map of the ground truth. The entire slice is shown on the left, whereas there is a zoomed-in version on the right. Each voxel is colored based on its expected value. The voxels with $\mathrm{E}[T_i|\mathbf{D}] < 0.05$ are not colored.

From one of the traces (Fig. 3), it can be seen that the sampler is rapidly exploring the parameter space and did not seem to have any convergence problem.

After sampling, a ground truth with soft labels was estimated with `get_ground_truth` (Fig. 4.) To obtain a ground truth with hardened labels, it is possible to set a threshold, e.g.:

```
binary_ground_truth = (soft_ground_truth >= 0.5).astype(int)
```

The uncertainty on the performance parameters was estimated with the 95% Highest Density Intervals (HDI):

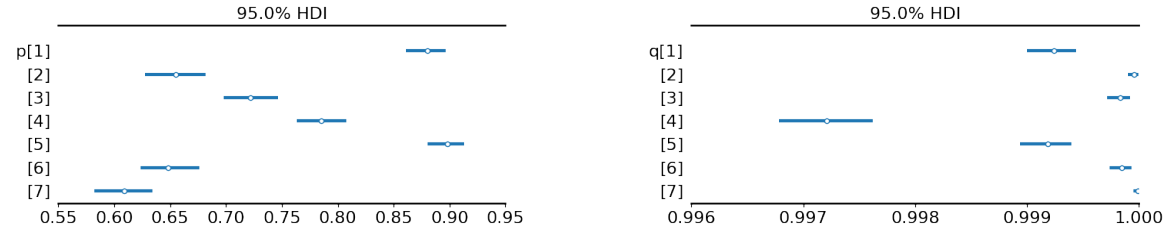
```
ax = az.plot_forest(
  sample,
  var_names=["p", "q"],
  combined=True, # combine the chains
  hdi_prob=0.95
)
```

The HDI plot can be seen in Fig. 5. As the intervals are wide, there is high uncertainty on the parameters and the ground truth could not be precise. For this reason, in section 4.5 more data was added to reduce the uncertainty.

The statistics of the performance parameters can be calculated by:

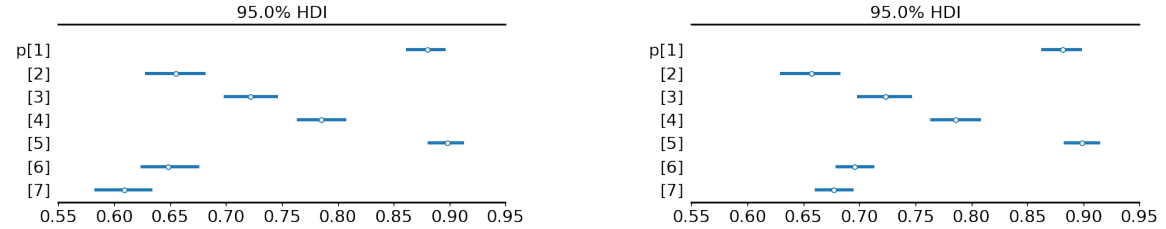
```
az.summary(sample, var_names=["p", "q"])
```

These statistics can be used to estimate the uncertainty and to make a point estimate for the parameters.



(a) (b)

Figure 5: 95% HDI for the performance parameters. (a) Sensitivity, (b) specificity. Each row refers to a rater. The dot in the center of each interval is the median.



(a) (b)

Figure 6: 95% HDI for the sensitivity with (a) and without (b) prior knowledge for raters 6 and 7.

4.3 Prior knowledge injection on raters' parameters

For this dataset, there was no prior knowledge available on the parameters. However, assuming that it was known that two raters correctly identified 1000 lesion voxels and misclassified 500 lesion voxels, this knowledge could have been inserted into the model in the following way:

```
model = BayesianSTAPLE(D, alpha_p = [1,1,1,1,1,1000,1000],
            beta_p=[1,1,1,1,1,500,500])
```

The more the prior is informative (i.e., the higher values for the hyperparameters) the more biased is the final estimate (Fig. 6).

If a rater had had perfect recall (sensitivity equal to one), the model could have been defined as:

```
max_v = np.finfo(np.float64).max
model = BayesianSTAPLE(D, alpha_p=[1,1,1,1,1,max_v,1])
```

In this case, all the lesions in the ground truth would have been also in the segmentation of rater 6.

4.4 Prior knowledge injection on the ground truth

For the voxels i that are known to be background, it is possible to fix the label by passing a matrix W such that $W_i = 0$. The same can be done for lesion voxels by setting $W_i = 1$. However, for this dataset, this information was not available.

Another way to insert prior knowledge is by specifying the number of background voxels and structure voxels that are expected to be in the ground truth:

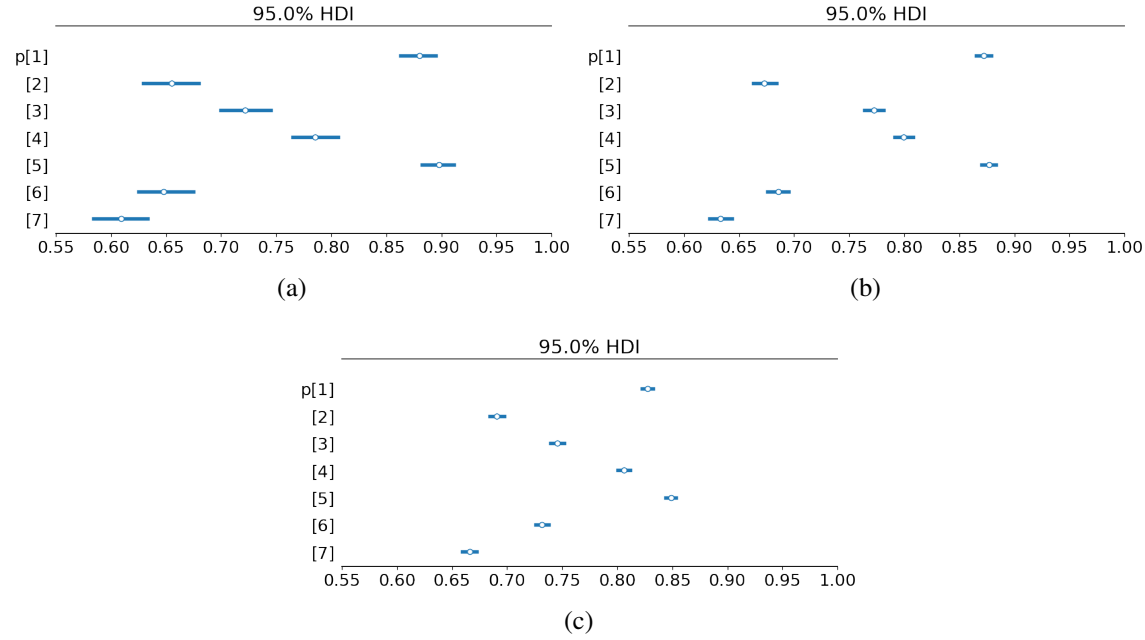


Figure 7: Posterior sensitivity with different number of contiguous slices. (a) 1 slice, (b) 5 slices and (c) 9 slices. The contiguous slices were taken before and after the one that was fixed.

```
num_bg_voxels = 50000 # background
num_struct_voxels = 1000 # structure
model = BayesianSTAPLE(D, alpha_w = num_struct_voxels,
            beta_w= num_bg_voxels)
```

Also, in this case, this information was not available.

4.5 Uncertainty reduction

To reduce the uncertainty, contiguous slices were added to the one being analyzed. As shown in Fig. 7, increasing the number of contiguous slices progressively reduces uncertainty, as expected. However, using very distant slices is not advised as the estimated performance could be altered and may not reflect the performance of the analyzed slice. In general, to avoid this problem, it is suggested that the raters segment the analyzed slice multiple times and use those data instead.

4.6 Performances on CPU and GPU

Bayesian STAPLE was executed both on the CPU and on the GPU, exponentially increasing the input size. As shown in Fig. 8, GPU execution is significantly faster with large inputs. The execution time with CPU increases exponentially.

5 Conclusion

In this work, a Python module was presented that implements the Bayesian version of STAPLE. This software improves the original version of STAPLE by an explicit representation of

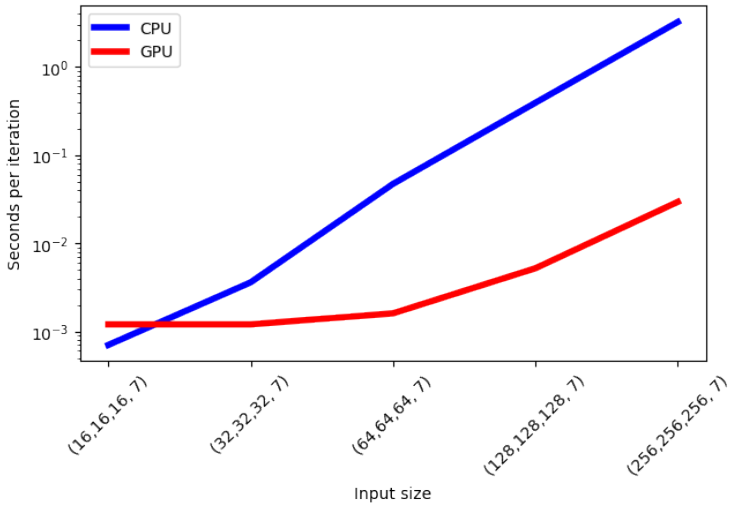


Figure 8: Execution time for each sampling iteration (in secs.). Original images were subsampled to test the algorithm on different input sizes.

uncertainty on both raters' performance and ground truth, thus providing richer information to help decision making. Also, prior knowledge can be introduced in the model, e.g. by using raters' performance estimated in previous assessment sessions. Furthermore, the information on the available data can be used to inject prior knowledge in the ground truth (e.g. voxels luminosity in medical images).

The experimental results showed how to use the software in the context of medical image segmentation. The resulting uncertainty can be conveniently represented as density intervals to allow a visual inspection of the model uncertainty. Moreover, the results demonstrated that the software performs very efficiently on GPUs, and the procedure scales well with input size.

The current version of our library has some limitations, which could be overcome in future developments. Specifically, it supports binary labels only, which may limit the application of the library in multi-class problems. Its extension to more than two labels is possible by replacing the current representation of sensitivity and specificity of each rater with a probabilistic representation of a confusion matrix, which requires the replacement of Beta distributions with Dirichlet distributions. Moreover the probabilistic representation of the ground truth would require a categorical distribution in place of the current representation with a Bernoulli distribution.

The current implementation of Bayesian STAPLE assumes that the data items are independent, which can be a strong assumption in some cases. In particular, in medical image segmentation, neighboring voxels are more likely to have the same label. Data dependency could be conveniently modeled through Markov Random Fields, which can be effectively integrated in future implementations of the software.

Finally, in the current implementation, the prior distributions are fixed. The software only allows Beta distributions for representing raters and ground truth; in case of complex multimodal prior knowledge, this may be limiting. A possible solution considers more general probability distributions; however, this could lead to drop the Gibbs sampling strategy, which requires closed-form conditional posteriors. It would be still possible to use general multimodal distributions to be sampled by MCMC-based approaches, but this may introduce some inefficiency issues. A trade-off between these two extremes could be the use of mixtures of Beta distributions, which would still preserve the possibility of using Gibbs sampling.

Acknowledgments

The authors are members of the INdAM research group GNCS.

Declaration of interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] A. Akhondi-Asl and S. K. Warfield. *Estimation of the Prior Distribution of Ground Truth in the STAPLE Algorithm: An Empirical Bayesian Approach*. *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2012*, vol. 7510, Berlin, Heidelberg: Springer Berlin Heidelberg. 593–600. URL http://dx.doi.org/10.1007/978-3-642-33415-3_73

[2] B. Audelan, D. Hamzaoui, S. Montagne, R. Renard-Penna, and H. Delingette. *Robust Bayesian Fusion of Continuous Segmentation Maps*. Medical Image Analysis, vol. 78, (2022), 102398. URL http://dx.doi.org/10.1016/j.media.2022.102398

[3] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang. *JAX: composable transformations of Python+NumPy programs* (2018). URL http://github.com/google/jax

[4] M. Brett, C. J. Markiewicz, M. Hanke, M.-A. Côté, B. Cipollini, P. McCarthy, D. Jarecka, C. P. Cheng, E. Larson, Y. O. Halchenko, M. Cottaar, S. Ghosh, D. Wassermann, S. Gerhard, G. R. Lee, Z. Baratz, H.-T. Wang, D. Papadopoulos Orfanos, E. Kastman, J. Kaczmarzyk, R. Guidotti, J. Daniel, O. Duek, A. Rokem, M. Scheltienne, C. Madison, A. Sólon, B. Moloney, F. C. Morency, M. Goncalves, R. Markello, C. Riddell, C. Burns, J. Millman, A. Gramfort, J. Leppäkangas, J. J. van den Bosch, R. D. Vincent, H. Braun, K. Subramaniam, A. Van, K. J. Gorgolewski, P. R. Raamana, J. Klug, R. Vos de Wael, B. N. Nichols, E. M. Baker, S. Hayashi, B. Pinsard, C. Haselgrove, M. Hymers, O. Esteban, S. Koudoro, F. Pérez-García, J. Dockès, N. N. Oosterhof, B. Amirbekian, H. Christian, I. Nimmo-Smith, L. Nguyen, P. Suter, S. Reddigari, S. St-Jean, E. Panfilov, E. Garyfallidis, G. Varoquaux, J. H. Legarreta, K. S. Hahn, L. Waller, O. P. Hinds, B. Fauber, B. Dewey, F. Perez, J. Roberts, J.-B. Poline, J. Stutters, K. Jordan, M. Cieslak, M. E. Moreno, T. Hrnčiar, V. Haenel, Y. Schwartz, B. C. Darwin, B. Thirion, C. Gauthier, I. Solovey, I. Gonzalez, J. Palasubramaniam, J. Lecher, K. Leinweber, K. Raktivan, M. Calábková, P. Fischer, P. Gervais, S. Gadde, T. Ballinger, T. Roos, V. R. Reddam, and freec84. *nipy/nibabel: 5.2.1* (2024). URL http://dx.doi.org/10.5281/zenodo.10714563

[5] D. Cazzorla and C. Mencar. *Uncertainty Estimation of Raters' Performance and Ground Truth Through a Bayesian Extension of STAPLE*. B. Moser, L. Fischer, A. Mashkoor, J. Sametinger, A.-C. Glock, M. Mayr, and S. Luftensteiner (eds.), *Database and Expert Systems Applications - DEXA 2024 Workshops*, vol. 2169, Springer Nature Switzerland. 91–101. URL http://dx.doi.org/10.1007/978-3-031-68302-2_8

[6] O. Commowick, A. Istace, M. Kain, B. Laurent, F. Leray, M. Simon, S. C. Pop, P. Girard, R. Améli, J.-C. Ferré, A. Kerbrat, T. Tourdias, F. Cervenansky, T. Glatard, J. Beaumont, S. Doyle, F. Forbes, J. Knight, A. Khademi, A. Mahbod, C. Wang, R. McKinley, F. Wagner, J. Muschelli, E. Sweeney, E. Roura, X. Lladó, M. M. Santos, W. P. Santos, A. G. Silva-Filho, X. Tomas-Fernandez, H. Urien, I. Bloch, S. Valverde, M. Cabezas, F. J. Vera-Olmos, N. Malpica, C. Guttmann, S. Vukusic, G. Edan, M. Dojat, M. Styner, S. K. Warfield, F. Cotton, and C. Barillot. *Objective Evaluation of Multiple Sclerosis Lesion Segmentation Using a Data Management and Processing Infrastructure*. vol. 8, 1, (2018), 13650. URL http://dx.doi.org/10.1038/s41598-018-31911-7

[7] O. Commowick, M. Kain, R. Casey, R. Ameli, J.-C. Ferré, A. Kerbrat, T. Tourdias, F. Cervenansky, S. Camarasu-Pop, T. Glatard, S. Vukusic, G. Edan, C. Barillot, M. Dojat, and F. Cotton. *Multiple sclerosis lesions segmentation from multiple experts: The MICCAI 2016 challenge dataset*. NeuroImage, vol. 244, (2021), 118589. URL http://dx.doi.org/10.1016/j.neuroimage.2021.118589

[8] O. Commowick and S. Warfield. *Estimation of Inferential Uncertainty in Assessing Expert Segmentation Performance From STAPLE*. IEEE Transactions on Medical Imaging, vol. 29, 3, (2010), 771–780. URL http://dx.doi.org/10.1109/TMI.2009.2036011

[9] H. Du, T. N. Bradbury, J. A. Lavner, A. L. Meltzer, J. K. McNulty, L. A. Neff, and B. R. Karney. *A comparison of Bayesian synthesis approaches for studies comparing two means: A tutorial*. Research Synthesis Methods, vol. 11, 1, (2020), 36–65. URL http://dx.doi.org/10.1002/jrsm.1365

[10] A. E. Gelfand. *Gibbs Sampling*. vol. 95, 452, (2000), 1300–1304. URL http://dx.doi.org/10.1080/01621459.2000.10474335

[11] J. G. Ibrahim, M.-H. Chen, Y. Gwon, and F. Chen. *The power prior: theory and applications*. Statistics in Medicine, vol. 34, 28, (2015), 3724–3749. URL http://dx.doi.org/10.1002/sim.6728

[12] L. Joskowicz, D. Cohen, N. Caplan, and J. Sosna. *Inter-observer variability of manual contour delineation of structures in CT*. European Radiology, vol. 29, 3, (2019), 1391–1399. URL http://dx.doi.org/10.1007/s00330-018-5695-5

[13] C. Koller, G. Kauermann, and X. X. Zhu. *Going Beyond One-Hot Encoding in Classification: Can Human Uncertainty Improve Model Performance in Earth Observation?* IEEE Transactions on Geoscience and Remote Sensing, vol. 62, (2024), 1–11. URL http://dx.doi.org/10.1109/TGRS.2023.3336357

[14] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy. *Learning From Crowds*. J. Mach. Learn. Res., vol. 11, (2010), 1297–1322. URL http://dx.doi.org/10.5555/1756006.1859894

[15] N. Robitaille and S. Duchesne. *Label Fusion Strategy Selection*. International Journal of Biomedical Imaging, vol. 2012, 1, (2012), 431095. URL http://dx.doi.org/10.1155/2012/431095

[16] A. Ruiz, O. Martinez, X. Binefa, and F. M. Sukno. *Fusion of Valence and Arousal Annotations through Dynamic Subjective Ordinal Modelling*. *2017 12th IEEE International*

Conference on Automatic Face & Gesture Recognition (FG 2017). IEEE, 331–338. URL http://dx.doi.org/10.1109/FG.2017.48

[17] L. Sun, C. Zu, W. Shao, J. Guang, D. Zhang, and M. Liu. *Reliability-based robust multi-atlas label fusion for brain MRI segmentation*. Artificial intelligence in medicine, vol. 96, (2019), 12–24. URL http://dx.doi.org/10.1016/J.ARTMED.2019.03.004

[18] K. Van Leemput and M. R. Sabuncu. *A Cautionary Analysis of STAPLE Using Direct Inference of Segmentation Truth. Medical Image Computing and Computer-Assisted Intervention – MICCAI 2014*, vol. 8673, Cham: Springer International Publishing. 398–406. URL http://dx.doi.org/10.1007/978-3-319-10404-1_50

[19] S. K. Warfield, K. H. Zou, and W. M. Wells. *Simultaneous truth and performance level estimation (STAPLE): an algorithm for the validation of image segmentation*. IEEE Transactions on Medical Imaging, vol. 23, 7, (2004), 903–921. URL http://dx.doi.org/10.1109/TMI.2004.828354

[20] D. Zhou, Q. Liu, J. C. Platt, and C. Meek. *Aggregating ordinal labels from crowds by minimax conditional entropy. Proceedings of the 31st International Conference on International Conference on Machine Learning*, *ICML'14*, vol. 32. JMLR.org, *ICML'14*, vol. 32, II–262–II–270. URL http://dx.doi.org/10.5555/3044805.3044922