IDIAP COMMUNICATION REPORT

A BAYESIAN APPROACH TO MACHINE LEARNING MODEL COMPARISON

Antonio Morais

Idiap-Com-01-2023

FEBRUARY 2023

Centre du Parc, Centre du Parc, Rue Marconi 19, CH - 1920 Martigny
T +41 27 721 77 11 F +41 27 721 77 12 info@idiap.ch www.idiap.ch

Master project

A Bayesian approach to machine learning model comparison

Student :
Antonio Morais

Supervisor:
André Anjos
Professor:
Jean-marc Odobez

February, 2022

1 Acknowledgments

I would like to thank my supervisor, Dr. André Anjos, for all the help and guidance he provided me. Thanks also to my family and friends for all the support they gave me during these last months.

In loving memory of Domingos Manuel Jesus Caeiro Godinho, a friend who departed this life too soon.

Contents

2 Abstract

Performance measures are an important component of machine learning algorithms. They are useful when it comes to evaluate the quality of a model, but also to help the algorithm improve itself. Every need has its own metric. However, when we have a small data set, these measures don't express properly the performance of the model. That's when confidence intervals and credible regions come in handy. Expressing the performance measures in a probabilistic setting lets us develop them as distributions. Then we can use those distributions to establish credible regions. In the first instance we will address the precision, recall and F1-score followed by the accuracy, specificity and Jaccard index. We will study the coverage of the credible regions computed through the posterior distributions. Then we will discuss ROC curve, precision-recall curve and k-fold cross-validation. Finally we will conclude with a small discussion about what we could do with dependent samples.

3 Introduction

Machine learning is a branch of computer science widely used in different fields nowadays. Machine learning algorithms uses data to train a model so that it can make decisions in future input data. It is notably useful to detect diseases in medical scans or to detect different objects and delimitation in self driving cars.

When using machine learning for binary classification tasks, we can define a status for the samples according to their predicted output and their real label. So, for example, if we want to detect pictures of cats and we hand over one to our algorithm that he, in fact, detect as a cat, then we have a true positive sample. Doing so, we will divide the sample's status into 4 cases: true positives, true negatives, false positives and false negatives. First, we should define what a positive sample is. It is the samples that the algorithm considers as fulfilling the condition, so in our cats' picture detection example it will be every image that the algorithm thinks as cats. In contrary every output that does not respect the condition according to the algorithm will be designated as negatives. Also a sample will be considered true (respectively false) if it is correctly (respectively incorrectly) predicted. Therefore we can divide the samples using the table 1.

		prediction (z)	
		positive (+)	negative (-)
label (l)	positive (+)	True positives (TP)	False negatives (FN)
	negative (-)	False positives (FP)	True negatives (TN)

Table 1: Group division of samples according to the output of the algorithm and their real label in a binary classification environment.

To evaluate algorithms we use different performance measurements. The accuracy of machine learning algorithms is one such measurement that encodes an intuitive meaning. Accuracy is the number of predictions correctly made by the model divided by the total number of predictions. At first accuracy seems to be a really good value to evaluate machine learning algorithms but it doesn't take every need into account. For example, if we want to do disease detection, it is more important to correctly detect sick patients than healthy people. For this kind of algorithm, we would prefer to compare their recall score, a score that returns the percentage of relevant items retrieved. As algorithms have different uses, different performance measures exist to answer every need.

However, a single value does not give a complete view on the performance of a system. In fact a machine learning algorithm depends on the data samples it uses and on different parameters so it could actually return a span of values for the performance measures. This span of values can be represented through confidence intervals and credible regions. These intervals give a better view on the performance of a system, which allows us to compare algorithms in a simpler way, notably when the data samples size is small meaning that the model is heavily dependent on the data.

In this thesis we will study the use of confidence intervals and credible regions for performance measurements. We will study several performance measures: precision, recall, specificity, accuracy, F1 score and Jaccard index. We will also look into ROC and precision-recall curves.

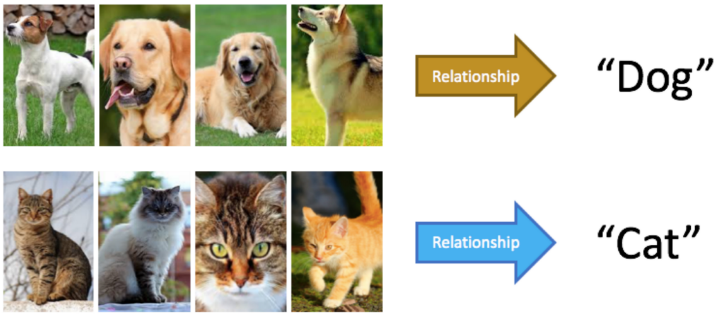


Figure 1: Example of images used in a classification machine learning algorithm. [1]

Usually when we compute the confidence interval of a performance measure for a machine learning algorithm, it is assumed that the data samples are independently and identically distributed (i.i.d.). We assume that in order to use formulas and algorithms designed especially for i.i.d. samples. Some machine learning algorithms indeed have i.i.d. samples. For example when we want to label pictures of animals, each of the images are i.i.d. samples (see Figure 1). However other algorithms use data that are not i.i.d. notably when the data samples are the pixels of the image. When this is the case pixels from the same image are not i.i.d. and often this type of algorithm, called semantic segmentation, uses the pixels to detect different objects in an image. For example machine learning used in the medical domain detects different parts in organ scans (see Figure 2).

When we compute confidence intervals, a level of confidence that we can choose is 95%, which means that we have a 95% probability that the true value of the performance measure lies inside the interval. We can test the performance of a method that returns

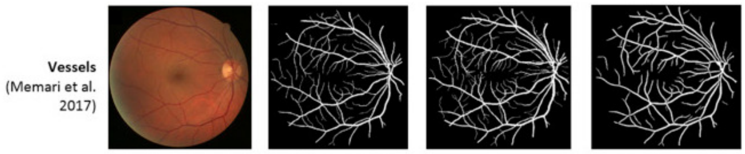


Figure 2: Example of an image from a retina photograph and a picture of the vessels present in the eye. [2]

confidence intervals by simulating systems where the real performance value is known and use the method on that system several times to compute the number of times the value is included in the interval. If the number of times the value is included is 95% then the method performs as desired. If this number is below 95% it means that the method returns an interval that is too restrictive and has more chances to not include the real value of the performance measure, in this case we consider that we have an optimistic view. In the contrary, the pessimistic view happens if this number is above 95% then the method returns an interval that is too big and doesn't give enough information on where lies the real value. Using confidence intervals to compare a performance measure of two machine learning models is pretty useful as it allows to compare a span of values rather than a single value.

Knowing how to measure confidence intervals when the samples are not i.i.d. on classical binary output machine learning models would help to have a better view on the performances of this kind of machine learning algorithms. Now we usually use the same algorithm whether the data is i.i.d. or not but these algorithms assume that the data is i.i.d. so the information retrieved from this confidence interval is biased. When using i.i.d. methods to compute the confidence interval for non i.i.d. samples we think that we should observe a confidence interval that is restrictive meaning the real value of the performance measure has more chances to be outside the bound. If that is the case this kind of algorithm would not fit for these uses as it would not perform as desired but it should be expected.

One idea in order to answer to the problem of the non i.i.d. samples would be to try to approximate the i.i.d. case with dependent samples. This hypothesis would choose pixels that are more independent from each other in a single sample and use them to compute the confidence interval. Doing so it might be possible to approximate the i.i.d. case and if that's what happens we would be able to use i.i.d. methods on dependent data without being afraid to receive bad information. The goal of this thesis is to study the use of credible region to compare systems from a Bayesian approach and we will also look into confidence interval for the frequentist approach.

4 State of the art

In this thesis we will talk about credible regions and confidence intervals. They are both intervals that give a span of values for performance measures in a similar way. One of the difference they have is that one uses the underlying distribution while the other uses an empirical distribution, respectively credible regions and confidence intervals. Indeed credible regions are computed using Bayesian statistics while confidence intervals are issued from frequentist statistics. Another point where they differ is their exact definition. The definition of the confidence intervals states that "95% of confidence intervals computed at the 95% confidence level contain the parameter"[3] meaning that if we compute 100 confidence intervals of a performance measure for a machine learning algorithm, then 95 of them should contain the true value of the measure we're computing. The credible region, on the other side, describes the probability that the true value of the parameter lies between the two bounds of the interval meaning that "if the subjective probability that μ lies between 35 and 45 is 0.95, then $35 \leq \mu \leq 45$ is a 95% credible interval" [4]. However, even with these differences, these two intervals are often used interchangeably and used to express the same thing. Results show that for linear models these intervals can be considered identical when several conditions are met, and for nonlinear models they can be considered identical if some assumptions are satisfied [5].

A frequentist approach to compute confidence intervals is bootstrapping. Bootstrapping is a technique that relies on random sampling with replacement. When doing this sampling we can infer different performance measures and properties of the underlying distribution in an empirical way [6][7]. This technique is pretty popular and whenever people want to compute confidence intervals, they most generally opt to bootstrapping to do so. Using this technique it is also possible to compute other measures for a machine learning algorithm, for example in Koch and Marshall [8] the authors use bootstrapping to obtain coverage plots and percentile intervals providing graphical information on the performance of the algorithms. Bootstrapping is also affected by data dependency and, according to Liu and Singh [9], when we construct bootstrap confidence intervals under the assumption that we are in an i.i.d. environment we can observe that these intervals become conservative when using non-i.i.d. models.

As discussed in the introduction, a single scalar for a performance measure often doesn't give enough information to the performance of the machine learning algorithms.

In fact these algorithms are heavily data dependent so obviously the performance measure will also depend on the data but as we never get the complete distribution because it would require infinite data we don't obtain the true value of the performance measure but a value that tends to approximate the real one the more data we have. A way to have a better view on this performance measure is to use confidence intervals, doing so we can use the dependence on the data from the algorithm to extend our view of the performance measure and it will include indirectly the data that we couldn't retrieve in the computation of the interval. Using confidence intervals to compare models is already done in the field of semantic segmentation and in other fields as it's an easy way to obtain more information on your algorithm, for example in Genc et al. [10] the authors compare three different algorithms in the segmentation of STEM tomography. In their comparison we can see that they used a 95% confidence interval to express the performance of each of the three algorithms. Another example would be in Sabogal et al. [11] where the authors present the results of their study with a 95% confidence interval for the cross-section measure. However confidence intervals are not the only way to give a better view on the performance of an algorithm. Indeed other papers, we will briefly review, try to explore new ways to express performance measure in order to obtain more information on the quality of the algorithms they are studying.

In Zhang et al. [12] the authors use, what they call, the micro-averaging and macro-averaging of the F1 scores in order to have a better view on this performance measure. Micro-averaging, as they describe it, uses the per-document predictions across classes in order to compute the micro-averaged recall and micro-averaged precision. After that they use those two values and compute their harmonic mean in order to obtain the micro-averaged F1 score. In the other hand they use each individual class to compute the precision and recall of each of them then they use these values to calculate the different F1 scores and when averaging those they obtain what they call the macro-averaging F1 score. Using these 2 values to express this performance measure allows them to better evaluate their machine learning algorithm and compare them in an environment giving more information.

Another paper by Yacouby and Axman [13] develops the precision, recall and F1 score in a new way with a probabilistic extension. Doing so the authors present new metrics that they refer as confidence-Precision, confidence-Recall and confidence-F1. These measures were mainly developed to help in natural language processing, as they answer to some of the challenges encountered in this field. When doing classification tasks in machine learning, the algorithm outputs probabilities to describe the chances to belong to the different classes. It is using these probabilities that they compute the new confidence metrics instead of using the true positives, true negatives, false positives and false negatives values. Doing so it gives a better vision of the performance of the algorithm, as we include information from samples in a non binary way. Indeed even if the algorithm outputs a class for a sample we will also use the information from the other classes for this sample. These new metrics have several benefits compared to their threshold based counterparts according to the authors, notably they pick up 4 main benefits. These advantages are the following: i) a better robustness to NaN values that are induced by a division by zero; ii) a sensitivity to changes in the model's confidence scores; iii) a lower variance inducing a better generalization to new data; iv) generally provide the same kind of ranking as the usual performance measures.

Extending performance measures are not the only way to gain information for different models. It is also possible to study existing metrics to detect which ones give the best information and in which conditions they do, indeed, succeed in their role. In Dinga et al. [14] for example, the authors do a comparison among four existing measures. These metrics are the accuracy, the area under the curve, the brier score and the logarithmic score. The 2 last being probabilistic performance measures whereas the 2 others don't really depend on probability. Ultimately they conclude that the accuracy is the worst measure among the four with respect to reliability of results, statistical power, selecting informative features and detecting model improvement. In the other hand the 2 probabilistic metrics are the ones that seem to perform the better while the area under the curve stands somewhere in between. That's why they advise to not use accuracy to make any statistical inference in order to improve the model performance but we should prefer other measures. Another reason they don't recommend the accuracy is because it weighs false positive and false negative misclassification equally but, in some contexts, this is not desired. Notably when standing in a medical perspective the results of a test or model might have a huge impact in the life of a patient inducing possible heavy operations while they were not necessary if he was a false positive.

These different extensions and studies assume one important point, indeed, in those papers, it is supposed that the data samples are i.i.d. which is often the case in most of the machine learning algorithms. However it is not always true, notably in semantic segmentation where the pixels of the images are often accounted as individual samples, therefore not being independent of each other. When we are in this situation using the formulas of the extensions described previously will probably yield results that are not completely truthful, as they are not suited to dependent samples. According to Wåsjø [15] having dependent samples induce biased error estimates. Indeed, in his study that focuses on wound detection, he divides the images into, what he calls, superpixel-edges and superpixel-segments that are mainly groups of pixels and are, therefore, samples that have dependencies between each other when they come from the same image. While training different algorithms, he detects that the biased error estimates induced by the dependent samples might cause suboptimal hyperparameters and features selections which, in turn, give a classifier that is less performant. So we know that if we have dependent samples, it definitely has an influence on the algorithm output and we should adapt the algorithm or the samples if we want to be able to use the different extensions discussed.

In Xiong et al. [16], authors managed to adapt the 3D convolutional neural network algorithm they were using to accommodate dependent samples and perform better than without any change. Training usually assumes independent samples and, according to them, even though this assumption is violated in the computer vision field, the millions of training examples compensate this and the empirical performance is satisfactory. Unfortunately, in semantic segmentation, studies are often done using only a few hundred images and, therefore, the assumption violation turns out to be more important and has more impact. Comparing their modified convolutional neural network to a baseline algorithm they managed to show empirical results proving that their changes bring an improvement to the performance of the algorithm. By showing that it was possible to adapt a 3D convolutional neural network algorithm in order to take the dependency in account, they revealed that it was probably possible to adapt other algorithms to also accommodate dependent samples.

5 Methods

In this chapter we describe a Bayesian approach by Goutte and Gaussier [17] for the credible region estimation[1]. First of all we define what a credible region is.

Definition 1 (Credible region or interval)**.** *A credible region for a parameter x is an interval defined by a lower bound L and an upper bound U such that the true value k of the parameter x lies between the two bounds with a probability α.*

$$P(k \in [L, U]) = \alpha$$

This means that, for example, if we compute the 95% credible region for the precision of a machine learning algorithm and we obtain the interval [0.82, 0.90] then we know that we have a probability of 95% that the real precision of this algorithm is between 0.82 and 0.90. The span of the credible region actually depends on several factors, the main one being the size of the data set used to evaluate the algorithm. These 2 sizes are inversely proportional from one to the other as the more data we have the more the algorithm will be complete and behave as the real underlying distribution. Bayesian statistics are at the center of credible regions, indeed, when computing a credible region, we assume a prior probability density function that models the likelihood of the parameter then, using Bayes theorem [19], we can establish the posterior distribution of the parameter given its current estimate.

As done in Goutte and Gaussier [17], we can redefine the precision and recall with a probabilistic definition in order to express them as estimates. Indeed, we know the formulas to compute the precision and recall:

$$\begin{aligned} precision = p &= \frac{TP}{TP + FP} \\ recall = r &= \frac{TP}{TP + FN} \end{aligned} \tag{1}$$

Looking more into this, we see that, actually, the precision returns the ratio of retrieved objects that are relevant while the recall returns the ratio of relevant objects returned

[1] All the code written that results from the following conclusions can be found in the measure module from the bob python package[18] accessible at https://gitlab.idiap.ch/bob/bob.measure also the graph generation code and other useful information in order to use the package can be found in the user guide.

by the system. With this in mind, probabilistic definitions that stand out are that the precision is "the probability that an object is relevant given that it is returned by the system"[17] and the recall is "the probability that a relevant object is returned"[17]. So, in mathematical terms, we can define them as (cf Table 1 for definition of l and z):

$$p = P(l = +|z = +)$$
$$r = P(z = +|l = +)$$

When observing the outputs of a machine learning algorithm, it divides all the samples into 4 distinct groups as discussed earlier. We can therefore represent the 4 groups as a multinomial distribution and the counts TP, TN, FP and FN as independent drawings from this multinomial. Therefore, like in Goutte and Gaussier [17], we can assume that:

Assumption 1. *$\mathcal{D} \equiv (TP, FP, FN, TN)$ is a multinomial distribution with parameters $\pi_{TP}, \pi_{FP}, \pi_{FN}$ and π_{TN}, $\mathcal{D} \sim \mathcal{M}(n; \pi)$ with $\pi \equiv (\pi_{TP}, \pi_{FP}, \pi_{FN}, \pi_{TN})$*

$$P(\mathcal{D} = (TP, FP, FN, TN)) = \frac{n!}{TP!FP!FN!TN!}\pi_{TP}^{TP}\pi_{FP}^{FP}\pi_{FN}^{FN}\pi_{TN}^{TN} \tag{2}$$

From the fact that this is a multinomial we can extract two main properties that will be useful for following proofs [17][20].

Property 1. *Each component i of $\mathcal{D}$ follows a binomial distribution $\mathcal{B}(n; \pi_i)$, with parameters n and identical probability π_i i.e. TP $\sim \mathcal{B}(n; \pi_{TP})$ etc.*

Property 2. *Each component i of $\mathcal{D}$ conditioned on another component j follows a binomial distribution $\mathcal{B}(n - n_j, \frac{\pi_i}{1-\pi_j})$ with parameters n - n_j and probability $\frac{\pi_i}{1-\pi_j}$ i.e. FN/TP $\sim \mathcal{B}(n - TP; \frac{\pi_{FN}}{1-\pi_{TP}})$ etc.*

In Goutte and Gaussier [17], authors proved that $p|\mathcal{D} \sim Be(TP + \lambda, FP + \lambda)$. In a similar fashion we will prove that $r|\mathcal{D} \sim Be(TP + \lambda, FN + \lambda)$.

First of all we should prove that TP+FN $\sim \mathcal{B}$(n, $\pi_{TP} + \pi_{FN}$)

Lemma 1. *TP+FN $\sim \mathcal{B}$(n, $\pi_{TP} + \pi_{FN}$)*

Proof. Using properties 1 and 2, we have TP $\sim \mathcal{B}(n; \pi_{TP})$ and FN|TP $\sim \mathcal{B}(n-TP; \frac{\pi_{FN}}{1-\pi_{TP}})$

so

$$\begin{aligned}
P(TP+FN=k) &= \sum_{x=0}^{k} P(TP=x)P(FN=k-x|TP=x) \\
&= \sum_{x=0}^{k} \binom{n}{x} \pi_{TP}^{x}(1-\pi_{TP})^{n-x} \binom{n-x}{k-x} \left(\frac{\pi_{FN}}{1-\pi_{TP}}\right)^{k-x} \left(1-\frac{\pi_{FN}}{1-\pi_{TP}}\right)^{n-k} \\
&= \sum_{x=0}^{k} \binom{n}{x}\binom{n-x}{k-x} \pi_{TP}^{x}\cancel{(1-\pi_{TP})^{n-x}} \frac{\pi_{FN}^{k-x}}{\cancel{(1-\pi_{TP})^{k-x}}} \frac{(1-\pi_{TP}-\pi_{FN})^{n-k}}{\cancel{(1-\pi_{TP})^{n-k}}} \\
&= \sum_{x=0}^{k} \binom{n}{x}\binom{n-x}{k-x} \pi_{TP}^{x}\pi_{FN}^{k-x} \left(1-\pi_{TP}-\pi_{FN}\right)^{n-k} \\
&= \sum_{x=0}^{k} \frac{k!}{k!}\frac{n!}{x!\cancel{(n-x)!}}\frac{\cancel{(n-x)!}}{(n-k)!(k-x)!} \pi_{TP}^{x}\pi_{FN}^{k-x} \left(1-\pi_{TP}-\pi_{FN}\right)^{n-k} \\
&= \binom{n}{k} \left(1-\pi_{TP}-\pi_{FN}\right)^{n-k} \sum_{x=0}^{k} \binom{k}{x} \pi_{TP}^{x}\pi_{FN}^{k-x} \\
&= \binom{n}{k} \left(1-(\pi_{TP}+\pi_{FN})\right)^{n-k} \left(\pi_{TP}+\pi_{FN}\right)^{k}
\end{aligned}$$

Which is indeed a binomial with parameters n and $\pi_{TP}+\pi_{FN}$. □

Then we have to prove that TP|TP+FN $\sim \mathcal{B}$(TP+FN, $\frac{\pi_{TP}}{\pi_{TP}+\pi_{FN}}$)

Lemma 2. *TP|TP+FN* $\sim \mathcal{B}$*(TP+FN,* $\frac{\pi_{TP}}{\pi_{TP}+\pi_{FN}}$*)*

Proof. Using properties 1 and 2, we have TP $\sim \mathcal{B}(n;\pi_{TP})$ and FN|TP $\sim \mathcal{B}(n-TP;\frac{\pi_{FN}}{1-\pi_{TP}})$.

From lemma 1 we have TP+FN $\sim \mathcal{B}$(n, $\pi_{TP} + \pi_{FN}$). We define M = TP + FN.

$$\begin{aligned}
P(TP = k|TP + FN = M) &= \frac{P(TP = k)P(FN = M - k|TP = k)}{P(TP + FN = M)} \\
&= \frac{\binom{n}{k}\pi_{TP}^{k}(1-\pi_{TP})^{n-k}\binom{n-k}{M-k}\left(\frac{\pi_{FN}}{1-\pi_{TP}}\right)^{M-k}\left(1-\frac{\pi_{FN}}{1-\pi_{TP}}\right)^{n-M}}{\binom{n}{M}(\pi_{TP}+\pi_{FN})^{M}(1-(\pi_{TP}+\pi_{FN}))^{n-M}} \\
&= \frac{\binom{n}{k}\binom{n-k}{M-k}}{\binom{n}{M}} \frac{\pi_{TP}^{k}(1-\pi_{TP})^{n-k}\frac{\pi_{FN}^{M-k}}{(1-\pi_{TP})^{M-k}}\frac{(1-\pi_{TP}-\pi_{FN})^{n-M}}{(1-\pi_{TP})^{n-M}}}{(\pi_{TP}+\pi_{FN})^{M}(1-(\pi_{TP}+\pi_{FN}))^{n-M}} \\
&= \frac{\frac{\cancel{n!}}{k!\cancel{(n-k)!}}\frac{\cancel{(n-k)!}}{(M-k)!\cancel{(n-M)!}}}{\frac{\cancel{n!}}{M!\cancel{(n-M)!}}} \\
&\frac{\pi_{TP}^{k}\cancel{(1-\pi_{TP})^{n-k}}\frac{\pi_{FN}^{M-k}}{\cancel{(1-\pi_{TP})^{M-k}}}\frac{\cancel{(1-\pi_{TP}-\pi_{FN})^{n-M}}}{\cancel{(1-\pi_{TP})^{n-M}}}}{(\pi_{TP}+\pi_{FN})^{M}\cancel{(1-(\pi_{TP}+\pi_{FN}))^{n-M}}} \\
&= \frac{M!}{k!(M-k)!}\frac{\pi_{TP}^{k}\pi_{FN}^{M-k}}{(\pi_{TP}+\pi_{FN})^{k}(\pi_{TP}+\pi_{FN})^{M-k}} \\
&= \binom{M}{k}\left(\frac{\pi_{TP}}{\pi_{TP}+\pi_{FN}}\right)^{k}\left(\frac{\pi_{FN}}{\pi_{TP}+\pi_{FN}}\right)^{M-k} \\
&= \binom{M}{k}\left(\frac{\pi_{TP}}{\pi_{TP}+\pi_{FN}}\right)^{k}\left(1-\frac{\pi_{TP}}{\pi_{TP}+\pi_{FN}}\right)^{M-k}
\end{aligned}$$

Which is indeed a binomial with parameters M and $\frac{\pi_{TP}}{\pi_{TP}+\pi_{FN}}$. □

We have seen before that we could redefine the recall in a probabilistic environment:

$$r = P(z = +|l = +) = \frac{P(z = +, l = +)}{P(l = +)}$$

We know that when the label and the prediction are positive it is true positive samples, therefore the probability P(z=+, l=+) = π_{TP}. If the label is positive then it is true positive and false negative samples, so the probability P(l=+) = $\pi_{TP} + \pi_{FN}$. We can then conclude that:

$$r = \frac{\pi_{TP}}{\pi_{TP} + \pi_{FN}}$$

So we can rewrite the binomial distribution of TP|TP+FN as $\mathcal{B}$(TP+FN, r).

The goal of all these reformulations is to interpret the recall as an estimate computed using the known values TP, FP, FN and TN from the algorithm. Indeed this reinterpretation will allow us to create a span of possible values that the algorithm could have returned with the information that we have on his predictions. In probabilistic terms what we're trying to compute is the posterior probability P(r$|\mathcal{D}$). As we know that TP|TP+FN $\sim \mathcal{B}$(TP+FN, r), it is possible to write the likelihood of r as:

$$\mathcal{L}(r) = P(\mathcal{D}|r) \propto r^{TP}(1-r)^{FN}$$

Using Bayes' rule[19], we can express the desired P(r$|\mathcal{D}$) as follows:

$$P(r|\mathcal{D}) \propto P(\mathcal{D}|r)P(r) \tag{3}$$

We use a symmetric beta distribution for the prior P(r) as we're dealing with binomials it is a logical choice.

$$r \sim \mathcal{B}e(\lambda, \lambda) : P(r) = \frac{\Gamma(2\lambda)}{\Gamma(\lambda)^2} r^{\lambda-1}(1-r)^{\lambda-1}$$

Consequently we can rewrite equation 3:

$$\begin{aligned} P(r|\mathcal{D}) &\propto r^{TP}(1-r)^{FN} r^{\lambda-1}(1-r)^{\lambda-1} \\ &\propto r^{TP+\lambda-1}(1-r)^{FN+\lambda-1} \end{aligned}$$

Which is a beta distribution with parameters TP + λ and FN + λ. Therefore r$|\mathcal{D} \sim$ Be(TP + λ, FN + λ) and, as seen in Goutte and Gaussier [17], $p|\mathcal{D} \sim Be(TP+\lambda, FP+\lambda)$. Using this information, we can compute credible regions for precision and recall by integrating these distributions[2].

To understand how useful the information provided by these distributions is, we will illustrate it by comparing 2 possible systems. Let's suppose that the first system has 10 true positive samples and 5 false negative ones and that the second system has 3 true positive and false negative samples. Using the usual formula 1, the recall of the first system would be $0.\overline{66}$ while the second one would be 0.5 so we could just conclude that system 1 outperforms the other one. But by looking at the graph from figure 3, we can deduce much more information. Indeed we see that system 1 is more consistent

[2] The measures function from the credible_region file returns credible regions for the precision and recall.

while system 2 is really sparse, almost ranging from 0 to 1 in the possible outcomes. However we can also see, using Monte Carlo simulation, that, in 24% of the cases, system 2 outperforms system 1 even if it seems unusual and this is probably due to how sparse it is. Monte Carlo is a method that uses random sampling to acquire numerical results for a simulation[21]. So, in this context, we will use the binomial distributions, that result from the given number of true positives and false negatives, to generate a lot of samples and use those to compare the 2 systems.

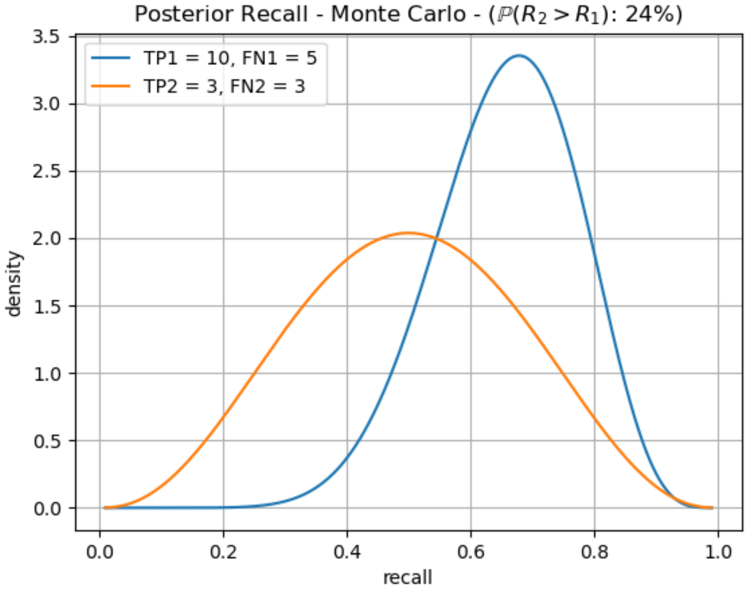


Figure 3: Recall distribution for 2 systems.

Now that we talked about precision and recall, we can talk about the F1-score. The usual formula to compute the F1-score uses the precision and recall.

$$F1 = \frac{2pr}{p + r} \tag{4}$$

To redefine the F1-score in a probabilistic perspective we will use 3 properties of gamma distributions as described in Goutte and Gaussier [17]. Let's suppose we have two gamma distributions with the same shape parameter. We will define them as X $\sim \Gamma(\alpha, h)$

and Y $\sim \Gamma(\beta, h)$. With this definition the three following properties hold.

$$\forall c > 0, cX \sim \Gamma(\alpha, ch) \tag{5}$$

$$X + Y \sim \Gamma(\alpha + \beta, h) \tag{6}$$

$$\frac{X}{X+Y} \sim Be(\alpha, \beta) \tag{7}$$

From property 7 and the beta definitions of precision and recall we can rewrite them as divisions and sums of gamma distributions. By defining three gamma distributions X $\sim \Gamma(TP + \lambda, h)$, Y $\sim \Gamma(FP + \lambda, h)$ and Z $\sim \Gamma(FN + \lambda, h)$ we obtain:

$$\begin{aligned} precision &= \frac{X}{X+Y} \\ recall &= \frac{X}{X+Z} \end{aligned}$$

Using those and the formula 4 we can rewrite the F1-score with gamma distributions.

$$\begin{aligned} F1 &= \frac{2\frac{X}{X+Y}\frac{X}{X+Z}}{\frac{X}{X+Y} + \frac{X}{X+Z}} \\ &= \frac{2\frac{XX}{(X+Y)(X+Z)}}{\frac{X(X+Z)+X(X+Y)}{(X+Y)(X+Z)}} \\ &= \frac{2X\cancel{X}\cancel{(X+Y)}\cancel{(X+Z)}}{\cancel{(X+Y)}\cancel{(X+Z)}\cancel{X}[(X+Y)+(X+Z)]} \\ &= \frac{2X}{2X+Y+Z} \\ &= \frac{\Gamma(TP+\lambda, 2h)}{\Gamma(TP+\lambda, 2h) + \Gamma(FP+FN+2\lambda, h)} \text{using 5 and 6} \end{aligned}$$

However, as these 2 gamma distributions don't have the same shape parameter, we can't rewrite the F1-score as a beta distribution using property 7 and we have to refer to Monte Carlo simulations in order to compare two systems and draw them in a graph. In figure 4, we can see the distribution generation produced by the Monte Carlo simulation. If we only computed the F1-score using the usual formula we would have 0.571 for the first system and 0.315 for the second one so we would conclude that system 1 is way better than system 2. The probabilistic outlook gives another point of view. Even though the first model seems preferable as it is more consistent and has a better mode and average, the second model doesn't seem so bad and actually outperforms system 1 in 43% of the cases.

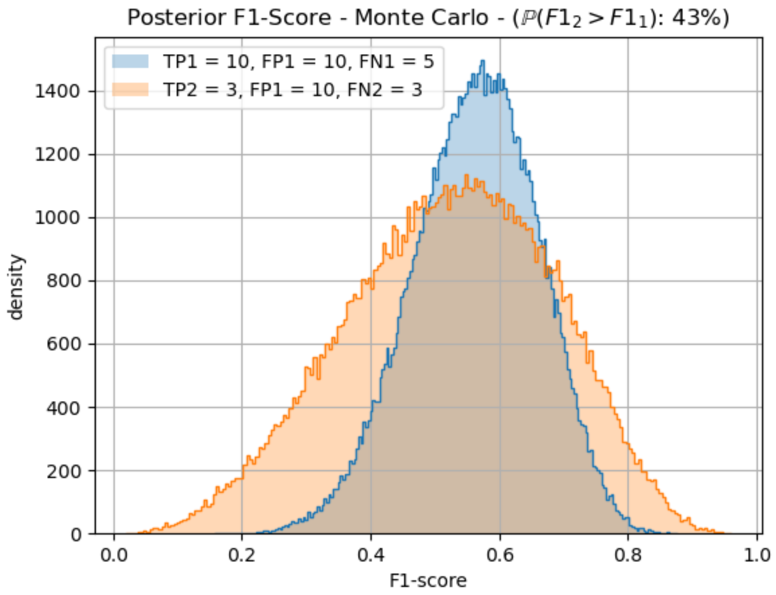


Figure 4: F1-score distribution for 2 systems using Monte Carlo simulations.

Monte Carlo simulations can also be used to compute the credible interval of the F1-score. Indeed using it we generate the empirical distribution of the Bayesian definition we found with the gamma distributions. This distribution then allows us to find the lower and upper values of the interval for the desired coverage[3].

Now that we have methods that return credible intervals for these performance measures we can try to evaluate them to see if they return the desired intervals. When evaluating this kind of methods, what we want to look into is the conservativeness of the method. We have 2 main outcomes for this evaluation. The first possibility is that the method is too conservative, meaning that it is pessimistic, and the credible regions returned are larger than the desired coverage. The other possibility is that the method is not conservative, therefore it is optimistic, and the interval returned is too restrictive compared to the desired coverage. In order to estimate conservativeness for our methods, we will rely on MMST [22]. In this blog, the author compares a credible region deduced from a binomial distribution, which is our case for precision and recall, to confidence intervals which are computed using frequentist approaches. So first we should define what

[3] The f1_score function in the credible region file returns this interval. The measures function discussed earlier also returns the credible interval of the F1-score among others.

confidence intervals are exactly.

Definition 2 (Confidence interval)**.** *For a parameter Θ, a x% confidence interval means that, when repeating the trials a large number of times, x% of the returned estimated intervals would include the true value of the parameter Θ.*

So the main difference between confidence interval and credible region is the following. For credible region, it is the parameter that is subject to random process whereas, for confidence interval, it is the interval that is subject to random process.

We will use three frequentist approaches and 2 Bayesian with different priors to make a comparison. The 2 different priors we will use are the flat prior, $\lambda = 1$, and the Jeffreys prior, $\lambda = 0.5$. The 3 confidence intervals[4] are methods that are based on binomial proportions[23] and they are:

1. The Wilson interval[24] is an improvement of the normal approximation which uses a normal distribution.
2. The Clopper–Pearson interval[25] which is referred as an exact method due to the use of the cumulative probabilities of the binomial distribution.
3. The Agresti-Coull interval[26] which is based on the Wald interval[27] and adds two successes and two failures. This means that we have to add 2 to the number of successes and 4 to the number of samples.

Using those intervals, we did a simulation to detect their conservativeness. In our simulation we know the exact value of the parameter we're trying to evaluate. Then we use the different methods presented before to compute the credible regions and confidence intervals for several systems created through the exact value of the parameter. Finally computing the percentage of times the exact value is included in the interval we can estimate the coverage of the different methods for all possible values of the parameter and deduce their conservativeness[5]. Doing so, we can generate a graph that shows the coverage of the 5 methods presented (see figure 5).

From this graph, we can detect several things. First we see that the Clopper-Pearson interval is too conservative. Indeed, we see that his coverage is always above the desired

[4] Implementations for these methods can be found in the confidence_interval file

[5] The code that generates this estimation is the function estimated_ci_coverage available in the curves file

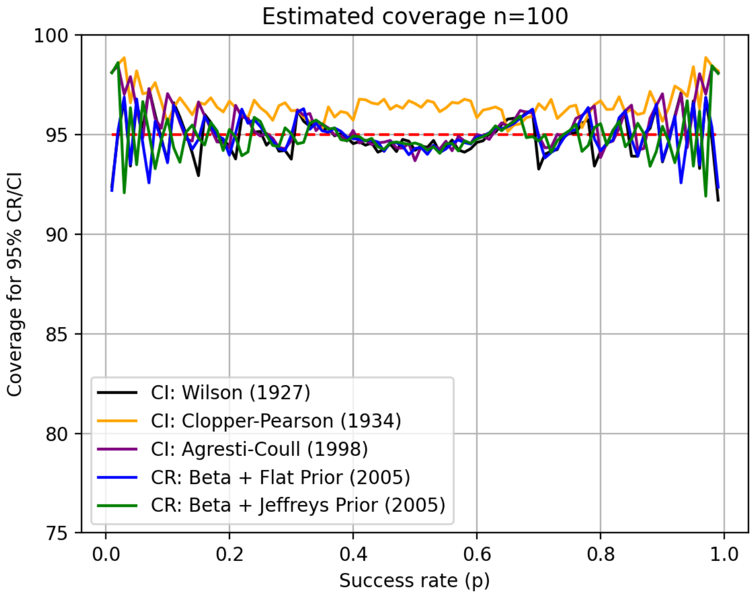


Figure 5: Coverage of the true parameter from 5 methods that return either a credible region or a confidence interval. This coverage was done using intervals at a 95% confidence level

95% confidence level. This was predictable as this method, as a consequence on how it's computed, often returns an interval containing the true value of the parameter even if it means that the interval will be bigger. The other 4 methods are quite similar in the coverage they have and we can detect some common features. When the true value lies between 0.35 and 0.65, the coverage is around 95% as it should. However, when the true value is around the limits, the coverage is much less predictable and can be too restrictive as too conservative. This is probably due to the fact that the binomial generation will vary much more when we are in those extreme cases. As we know that the precision and recall are similar to beta distributions, then the credible region returned for these parameters will behave as the beta priors presented in the figure 5. This will not be the case for the F1-score as we can't write it as a beta distribution.

6 Contributions

In this chapter, we will study the coverage for the F1-score and extend what we have seen previously in order to produce credible regions coming from other performance measures. As a first step, we will present the coverage of the F1-score then we will discuss the specificity, accuracy and Jaccard index metrics. After that we will talk about the receiver operating characteristic curve which allows to have a visualization of the confidence interval. In the end, we will discuss k-fold cross-validation for our credible regions.

As evoked earlier, the F1-score can not be represented as a beta distribution, therefore the figure 5 doesn't illustrate the coverage of the method we have to compute the credible region. To show the coverage, we had to design another function that uses Monte Carlo simulation. We consider that both precision and recall are issued from binomial distributions as we established earlier. So we will pass through possible values for the precision and recall. From these values we can compute the true value of the F1-score and using the binomial distributions we will generate several systems where the three true values of the metrics are, therefore, known. From the precision and recall generated with binomial distribution in each system we can compute the TP, FP and FN values. Then we can compute the credible region for the F1-score using our method and detect if the true value is included in the interval for each system. It is worthy to note that not every pair of precision and recall is possible as it would need more negative samples in some cases. The graph illustrating the F1-score coverage can be seen in figure 6.

As the F1-score depends on precision and recall, the graph is generated in 3D with the 2 metrics in the x and y axes and the coverage percentage in the z axis. We also have a second illustration that shows the 3D graph seen from above. We can see from the graph that the coverage of the F1-score is similar to the ones from the figure 5. Indeed, we see that the coverage is usually as desired except when we approach to the bounds of the precision and recall where they deteriorate[6].

Now we will extend the representation as beta distributions, seen in the previous chapter, to other performance measures. The metrics we will discuss are the specificity, the accuracy and the Jaccard index. The usual formulas to compute these measures using the TP, TN, FP and FN samples are the following:

[6] The function to generate this graph can be found in the doc/examples/f1-coverage file.

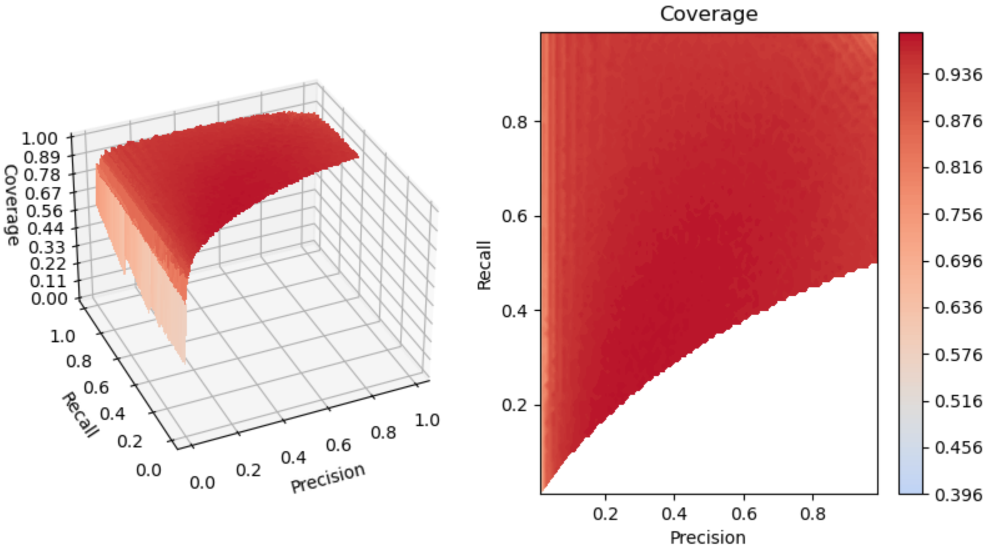


Figure 6: Coverage of the true F1-score from our method to compute the credible region. This coverage was done using an interval at a 95% confidence level

$$
\begin{aligned}
\text{specificity} = \text{true negative rate (TNR)} &= \frac{TN}{TN+FP} \qquad (8)\\
\text{accuracy} = acc &= \frac{TP+TN}{TP+TN+FP+FN}\\
\text{Jaccard index} = J_{index} &= \frac{TP}{TP+FN+FP}
\end{aligned}
$$

In a similar way than in the previous chapter, we can prove that these metrics are actually binomials and we can describe their posterior probability as beta distributions. More precisely, the specificity, which is the same as the true negative rate (TNR), has the following posterior TNR|D $\sim$ Be(TN + λ, FP + λ). The accuracy posterior is ACC|D $\sim$ Be(TP + TN + λ, FN + FP + λ) and the Jaccard index posterior is J_{index}|D $\sim$ Be(TP + λ, FN + FP + λ)[7]. We will now prove these statements.

Using properties 1, 2 and with similar proofs than lemmas 1 and 2, we can conclude 2 generalizations:

1. An addition of any two components from the multinomial results in a binomial (i.e.

[7] The measures function from the credible_region file also returns the credible regions for these metrics.

FN+FP $\sim \mathcal{B}$(n, $\pi_{FN} + \pi_{FP}$) etc.)

2. A component conditioned on an addition results on a binomial (i.e. FN|FN+FP $\sim$ $\mathcal{B}$(FN+FP, $\frac{\pi_{FN}}{\pi_{FP}+\pi_{FN}}$) etc.)

Lemma 3. *TNR/D $\sim$ Be(TN + λ, FP + λ)*

Proof. From the generalization 2, we have TN|TN+FP $\sim \mathcal{B}$(TN+FP, $\frac{\pi_{TN}}{\pi_{TN}+\pi_{FP}}$). Therefore the likelihood of TNR is :

$$\mathcal{L}(TNR) = P(\mathcal{D}|TNR) \propto TNR^{TN}(1 - TNR)^{FP}$$

Using Bayes' rule :

$$P(TNR|\mathcal{D}) \propto P(\mathcal{D}|TNR)P(TNR)$$

We can use the symmetric beta distribution for the prior distribution P(TNR).

$$TNR \sim \mathcal{B}e(\lambda, \lambda) : P(TNR) = \frac{\Gamma(2\lambda)}{\Gamma(\lambda)^2} TNR^{\lambda-1}(1 - TNR)^{\lambda-1}$$

So

$$\begin{aligned} P(TNR|\mathcal{D}) &\propto TNR^{TN}(1 - TNR)^{FP} TNR^{\lambda-1}(1 - TNR)^{\lambda-1} \\ &\propto TNR^{TN+\lambda-1}(1 - TNR)^{FP+\lambda-1} \end{aligned}$$

Which is a beta distribution with parameters TN + λ and FP + λ. We could conclude the same for the true positive rate, the false positive rate and the false negative rate, respectively TPR|D $\sim$ Be(TP + λ, FN + λ), FPR|D $\sim$ Be(FP + λ, TN + λ) and FNR|D $\sim$ Be(FN + λ, TP + λ). □

Lemma 4. *ACC/D $\sim$ Be(TP + TN + λ, FN + FP + λ)*

Proof. According to the generalization 1, we have TP+TN $\sim \mathcal{B}$(n, $\pi_{TP} + \pi_{TN}$). As n = TP + TN + FN + FP, we can actually rewrite this as TP+TN|TP+TN+FN+FP $\sim \mathcal{B}$(n, $\pi_{TP} + \pi_{TN}$). So the likelihood of ACC is :

$$\mathcal{L}(ACC) = P(\mathcal{D}|ACC) \propto ACC^{TP+TN}(1{-}ACC)^{1-(TP+TN)} = ACC^{TP+TN}(1{-}ACC)^{FN+FP}$$

Using Bayes' rule :

$$P(ACC|\mathcal{D}) \propto P(\mathcal{D}|ACC)P(ACC)$$

We can use the symmetric beta distribution for the prior distribution P(ACC).

$$ACC \sim \mathcal{B}e(\lambda, \lambda) : P(ACC) = \frac{\Gamma(2\lambda)}{\Gamma(\lambda)^2} ACC^{\lambda-1}(1 - ACC)^{\lambda-1}$$

So

$$\begin{aligned} P(ACC|\mathcal{D}) &\propto ACC^{TP+TN}(1 - ACC)^{FN+FP} ACC^{\lambda-1}(1 - ACC)^{\lambda-1} \\ &\propto ACC^{TP+TN+\lambda-1}(1 - ACC)^{FN+FP+\lambda-1} \end{aligned}$$

Which is a beta distribution with parameters TP + TN + λ and FN + FP + λ. □

Lemma 5. *TP+FN+FP* $\sim \mathcal{B}$*(n,* $\pi_{TP} + \pi_{FP} + \pi_{FN}$*)*

Proof. Using property 1, we know that TN $\sim \mathcal{B}(n; \pi_{TN})$ so

$$\begin{aligned} &P(TP + FN + FP = k) \\ &= P(TN = n - k) \\ &= \binom{n}{n-k} \pi_{TN}^{n-k} \left(1 - \pi_{TN}\right)^k \\ &= \binom{n}{k} \left(1 - (\pi_{TP} + \pi_{FP} + \pi_{FN})\right)^{n-k} \left(1 - (1 - (\pi_{TP} + \pi_{FP} + \pi_{FN}))\right)^k \\ &= \binom{n}{k} \left(1 - (\pi_{TP} + \pi_{FP} + \pi_{FN})\right)^{n-k} \left(\pi_{TP} + \pi_{FP} + \pi_{FN}\right)^k \end{aligned}$$

Which is indeed a binomial with parameters n and $\pi_{TP} + \pi_{FP} + \pi_{FN}$. □

Lemma 6. *TP/TP+FN+FP* $\sim \mathcal{B}$*(TP+FN+FP,* $\frac{\pi_{TP}}{\pi_{TP}+\pi_{FP}+\pi_{FN}}$*)*

Proof. From lemma 1 we can also conclude that FP+FN $\sim \mathcal{B}$(n, $\pi_{FP} + \pi_{FN}$). Using property 2, we can then conclude that FP+FN|TP $\sim \mathcal{B}(n - TP; \frac{\pi_{FP}+\pi_{FN}}{1-\pi_{TP}})$ so

$$\begin{aligned}
&P(TP = k|TP + FN + FP = M) \\
&= \frac{P(TP = k)P(FN + FP = M - k|TP = k)}{P(TP + FN + FP = M)} \\
&= \frac{\binom{n}{k}\pi_{TP}^k(1-\pi_{TP})^{n-k}\binom{n-k}{M-k}\left(\frac{\pi_{FP}+\pi_{FN}}{1-\pi_{TP}}\right)^{M-k}\left(1-\frac{\pi_{FP}+\pi_{FN}}{1-\pi_{TP}}\right)^{n-M}}{\binom{n}{M}(\pi_{TP}+\pi_{FP}+\pi_{FN})^M(1-(\pi_{TP}+\pi_{FP}+\pi_{FN}))^{n-M}} \\
&= \frac{\binom{n}{k}\binom{n-k}{M-k}}{\binom{n}{M}} \frac{\pi_{TP}^k(1-\pi_{TP})^{n-k}\frac{(\pi_{FP}+\pi_{FN})^{M-k}}{(1-\pi_{TP})^{M-k}}\frac{(1-\pi_{TP}-\pi_{FP}-\pi_{FN})^{n-M}}{(1-\pi_{TP})^{n-M}}}{(\pi_{TP}+\pi_{FP}+\pi_{FN})^M(1-(\pi_{TP}+\pi_{FP}+\pi_{FN}))^{n-M}} \\
&= \frac{\frac{\cancel{n}!}{k!\cancel{(n-k)}!}\frac{\cancel{(n-k)!}}{(M-k)!\cancel{(n-M)}!}}{\frac{\cancel{n}!}{M!\cancel{(n-M)}!}} \frac{\pi_{TP}^k\cancel{(1-\pi_{TP})^{n-k}}\frac{(\pi_{FP}+\pi_{FN})^{M-k}}{\cancel{(1-\pi_{TP})^{M-k}}}\cancel{\frac{(1-\pi_{TP}-\pi_{FP}-\pi_{FN})^{n-M}}{(1-\pi_{TP})^{n-M}}}}{(\pi_{TP}+\pi_{FP}+\pi_{FN})^M\cancel{(1-(\pi_{TP}+\pi_{FP}+\pi_{FN}))^{n-M}}} \\
&= \frac{M!}{k!(M-k)!}\frac{\pi_{TP}^k(\pi_{FP}+\pi_{FN})^{M-k}}{(\pi_{TP}+\pi_{FP}+\pi_{FN})^k(\pi_{TP}+\pi_{FP}+\pi_{FN})^{M-k}} \\
&= \binom{M}{k}\left(\frac{\pi_{TP}}{\pi_{TP}+\pi_{FP}+\pi_{FN}}\right)^k\left(\frac{\pi_{FP}+\pi_{FN}}{\pi_{TP}+\pi_{FP}+\pi_{FN}}\right)^{M-k} \\
&= \binom{M}{k}\left(\frac{\pi_{TP}}{\pi_{TP}+\pi_{FP}+\pi_{FN}}\right)^k\left(1-\frac{\pi_{TP}}{\pi_{TP}+\pi_{FP}+\pi_{FN}}\right)^{M-k}
\end{aligned}$$

Which is indeed a binomial with parameters TP+FN+FP and $\frac{\pi_{TP}}{\pi_{TP}+\pi_{FP}+\pi_{FN}}$. □

Lemma 7. *$J_{index}/D \sim \mathcal{B}e(TP + \lambda,\ FN + FP + \lambda)$*

Proof. From lemma 6, we know that TP|TP+FN+FP $\sim \mathcal{B}$(TP+FN+FP, $\frac{\pi_{TP}}{\pi_{TP}+\pi_{FP}+\pi_{FN}}$). Meaning that the likelihood of J_{index} is :

$$\mathcal{L}(J_{index}) = P(\mathcal{D}|J_{index}) \propto J_{index}^{TP}(1-J_{index})^{FN+FP} = J_{index}^{TP}(1-J_{index})^{FN+FP}$$

Using Bayes' rule :

$$P(J_{index}|\mathcal{D}) \propto P(\mathcal{D}|J_{index})P(J_{index})$$

We can use the symmetric beta distribution for the prior distribution P(J_{index}).

$$J_{index} \sim \mathcal{B}e(\lambda,\lambda) : P(J_{index}) = \frac{\Gamma(2\lambda)}{\Gamma(\lambda)^2}J_{index}^{\lambda-1}(1-J_{index})^{\lambda-1}$$

So

$$\begin{aligned} P(J_{index}|\mathcal{D}) &\propto J_{index}^{TP}(1 - J_{index})^{FN+FP} J_{index}^{\lambda-1}(1 - J_{index})^{\lambda-1} \\ &\propto J_{index}^{TP+\lambda-1}(1 - J_{index})^{FN+FP+\lambda-1} \end{aligned}$$

Which is a beta distribution with parameters TP + λ and FN + FP + λ. □

As we proved that the posterior of these metrics follow beta distributions, the coverage of their credible regions will be the same as the beta priors discussed in figure 5 depending if we use the flat prior or the Jeffreys' one.

In order to have a visualization of the confidence interval or credible region, the receiver operating characteristic curve (ROC curve) is a good way to achieve it. The machine learning algorithms in binary classification tasks don't return the class (positive or negative) directly for a sample. They return a probability that the sample is positive and it is the user of the algorithm that will decide of a threshold value according to which kind of samples (true positives, true negatives etc.) he wants to give more importance. Therefore the same algorithm outputs different numbers of TP, TN, etc., depending on this threshold value. The idea of the ROC curve is to pass through possible values of the threshold and compute the true positive rate (TPR) and false positive rate (FPR) at each of them. Then we display the computed values in a graph with the TPR and FPR as axes. As each threshold gives a different TPR and FPR, we will have a unique coordinate in the graph for each calculation. And with all these coordinates we can compute the ROC curve, but we want to display the interval too. So at each of these points we also compute the confidence interval (or the credible region depending on what algorithm we use, but from now on we will just refer to it as confidence interval for readability purposes) for the TPR and FPR. The upper and lower bounds that will be returned will produce new coordinates that will determine the limits of the interval in the graph. Then after doing all these we are able to display a ROC curve with a confidence interval[8]. From the ROC curve we can compute a new performance measure which is the area under the ROC curve (AUROC). This metric, as stated by the name, is the area that we can compute below the generated ROC curve. When computing the ROC curve with a confidence interval, we actually generate 2 more curves, an upper and a lower one, from which we can also compute the AUROC. Doing so, we also create a confidence interval for the AUROC.

[8] The roc_ci function in the curves file computes the ROC curve and its confidence interval

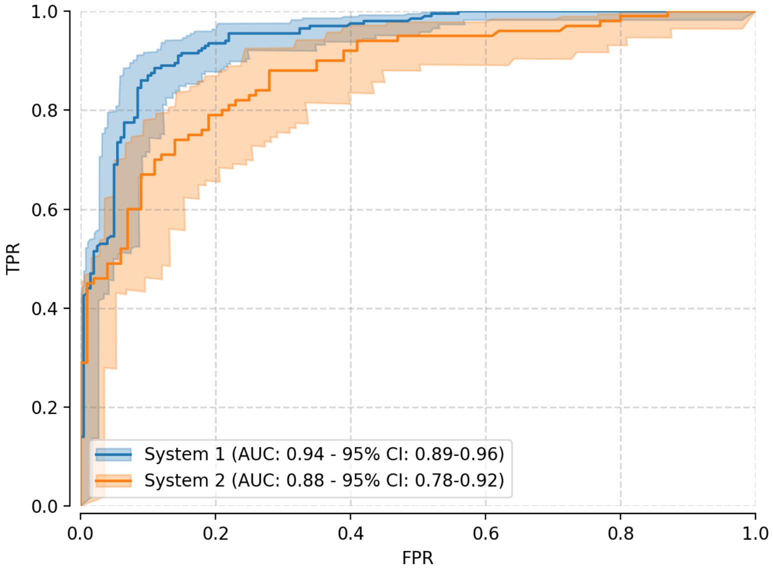


Figure 7: 2 ROC curves with a 95% credible region computed using a flat prior.

In figure 7, we can see 2 ROC curves generated from 2 systems. System 1 seems to perform better than the second one. It shows more consistency, as the confidence interval is narrower, and it has greater AUROC values (named just area under the curve in the figure) than its counterpart.

Using the same logic, as for the ROC curve, we can create a similar curve with the precision and recall instead of the rates used earlier. Doing so, we produce a precision-recall curve which also displays its confidence interval[9]. In the same manner than the AUROC, we can also define the area under the precision-recall curve (AUPRC). In figure 8, we see 2 systems being compared according to their precision-recall curves. One more time, system 1 seems to perform better, but it isn't as obvious than with the ROC curves, as there is more overlapping. However, system 1 seems more consistent, as we can see with the AUPRC confidence interval, which is narrower.

Lastly, we will present the k-fold cross-validation we developed to compute the credible region coming from several splits. The method we used is also known as Monte Carlo cross-validation[28]. The idea of this method is to average the results we obtained from each given split. The data division will be done from the user and he will input the TP, TN, FP and FN values from each split which he got after training the model. Once we

[9] The precision_recall_ci function in the curves file computes the precision-recall curve and its confidence interval

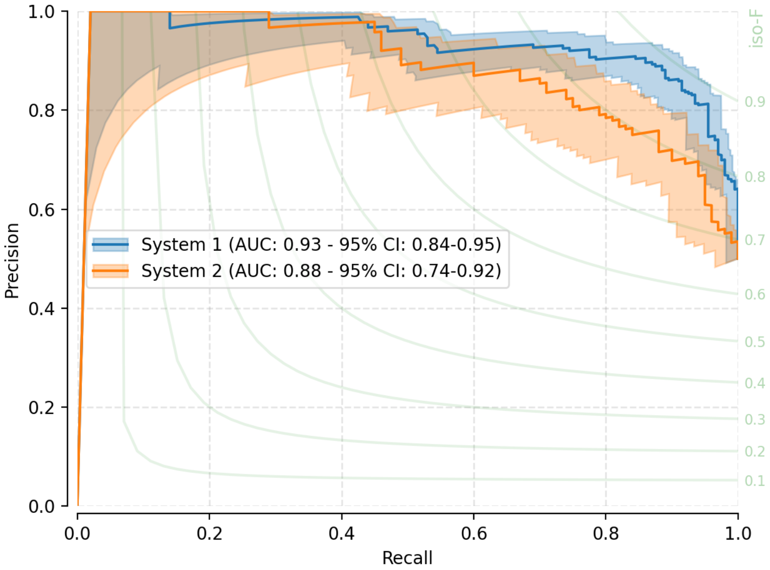


Figure 8: 2 precision-recall curves with a 95% credible region computed using a flat prior. The graph also includes iso-F1 lines, which are segments where we have the same F1-score value.

got all these values for each split, we know how to compute the posterior for one split depending on the desired measure (either a beta posterior for the binomial metrics or the posterior generated from Monte Carlo simulation for the F1-score). After computing the posteriors of every given split, we average them. Then we determine the credible region from the average posterior[10].

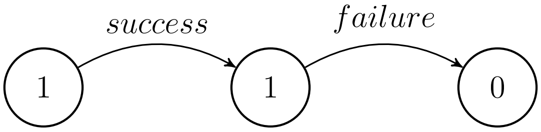
[10] The average_measures function in the credible_region file returns the k-fold credible region for all the metrics we presented. If we want to compute the k-fold credible region for only one measure we should either use the average_beta or average_f1score function with the correct parameters depending on the desired metric.

7 Conclusion

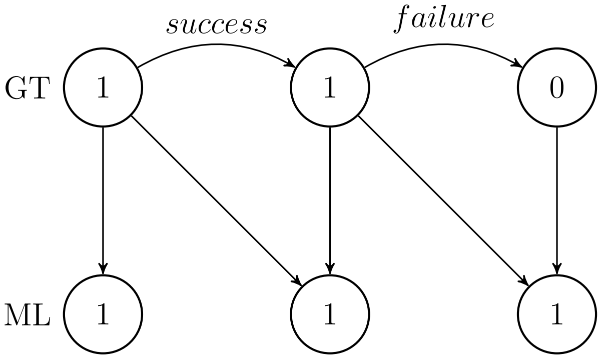
In conclusion of this thesis, we explored different ways to express credible regions and confidence intervals for useful performance measures. Doing so, we managed to compare different systems with information that was not accessible from just the unique value produced by the possible metrics. This information gave us a better view on the systems studied and it showed us that confidence intervals and credible regions were really useful, notably when the size of the data set is small. We also noticed that both frequentist and Bayesian approaches gave similar results and that they could be used analogously. However, it is also important to remark that they don't express the same result. So we should use them accordingly to what we're trying to convey.

In our contribution, we extended the work of Goutte and Gaussier [17] to include other performance measures like accuracy. We also completed the study of the F1-score by analyzing the coverage it produced and did the same for the other metrics, which could be expressed as binomials, according to the work of MMST [22]. After that we widened the ROC and precision-recall curves so that they also display the confidence interval or credible region of these measures. Doing so, it constructed intervals for the area under the curve too. In the end, we developed a function to express confidence intervals or credible regions when we employ k-fold cross-validation for our model.

An idea to continue this study on confidence intervals and credible regions would be to investigate the case when we have non i.i.d. samples, more precisely when there exists a dependency between them. To simulate dependent samples, we could rely on Markov chain Monte Carlo methods[29] and doing something similar to the Metropolis–Hastings algorithm[30][31]. This would divide the simulation into 2 steps. The first one would be to generate the ground truth that would be composed with positives (ones) and negatives (zeros). This could be done by making each sample depending to the previous one with a predetermined probability. Meaning that, for each sample, we would do a Bernoulli trial. On success the sample would take the same value as the previous one, and on failure it would be the opposite than the previous. As the first sample doesn't have a previous, it would do a Bernoulli trial with a probability unique to him to decide if it is a positive or a negative. The ground truth would look like that but with more samples:



The second step would be to generate machine learning outputs from this ground truth. In order to create dependency in the samples of the machine learning output, we have to make it dependent to the previous and his equivalent of the ground truth. Logically, the first sample would only depend on the first one of the ground truth. The dependencies would look like the following:



References

[1] Branav Kumar Gnanamoorthy. Machine Learning, October 2018. URL `https://medium.com/@gnabr/machine-learning-c28daf3cf60a`.

[2] Ursula Schmidt-Erfurth, Amir Sadeghipour, Bianca S. Gerendas, Sebastian M. Waldstein, and Hrvoje Bogunović. Artificial intelligence in retina. *Progress in Retinal and Eye Research*, 67:1–29, November 2018. ISSN 1350-9462. doi: 10.1016/j.preteyeres.2018.07.004. URL `https://www.sciencedirect.com/science/article/pii/S1350946218300119`.

[3] Confidence interval, February 2022. URL `https://en.wikipedia.org/w/index.php?title=Confidence_interval&oldid=1069395510`. Page Version ID: 1069395510.

[4] Credible interval, September 2020. URL `https://en.wikipedia.org/w/index.php?title=Credible_interval&oldid=976196216`. Page Version ID: 976196216.

[5] Dan Lu, Ming Ye, and Mary C. Hill. Analysis of regression confidence intervals and Bayesian credible intervals for uncertainty quantification. *Water Resources Research*, 48(9), 2012. ISSN 1944-7973. doi: 10.1029/2011WR011289. URL `https://onlinelibrary.wiley.com/doi/abs/10.1029/2011WR011289`.

[6] Michael R. Chernick. *Bootstrap methods: a guide for practitioners and researchers.* Wiley series in probability and statistics. Wiley-Interscience, Hoboken, N.J, 2nd ed edition, 2008. ISBN 978-0-471-75621-7. OCLC: ocn156785095.

[7] Robert J. Tibshirani Bradley Efron. *An Introduction to the Bootstrap.* Number 57 in Monographs on Statistics and Applied Probability. Chapman \& Hall/CRC, 1993.

[8] I. Koch and G. Marshall. Bootstrap coverage plots for image segmentation. In *Proceedings of 13th International Conference on Pattern Recognition*, pages 447–451 vol.2, Vienna, Austria, 1996. IEEE. ISBN 978-0-8186-7282-8. doi: 10.1109/ICPR.1996.546865. URL `http://ieeexplore.ieee.org/document/546865/`.

[9] Regina Y. Liu and Kesar Singh. Using i.i.d. bootstrap inference for general non-i.i.d. models. *Journal of Statistical Planning and Inference*, 43(1-2):67–75, January 1995. ISSN 03783758. doi: 10.1016/0378-3758(94)00008-J. URL `https://linkinghub.elsevier.com/retrieve/pii/037837589400008J`.

[10] Arda Genc, Libor Kovarik, and Hamish L. Fraser. A Deep Learning Approach for Semantic Segmentation of Unbalanced Data in Electron Tomography of Catalytic Materials. *arXiv:2201.07342 [cond-mat]*, January 2022. URL `http://arxiv.org/abs/2201.07342`. arXiv: 2201.07342.

[11] Sebastian Sabogal, Alan George, and Gary Crum. Reconfigurable Framework for Resilient Semantic Segmentation for Space Applications. *ACM Transactions on Reconfigurable Technology and Systems*, 14(4):1–32, December 2021. ISSN 1936-7406, 1936-7414. doi: 10.1145/3472770. URL `https://dl.acm.org/doi/10.1145/3472770`.

[12] Dell Zhang, Jun Wang, Xiaoxue Zhao, and Xiaoling Wang. A Bayesian Hierarchical Model for Comparing Average F1 Scores. URL `https://eprints.bbk.ac.uk/id/eprint/13086/1/PID3868347.pdf`.

[13] Reda Yacouby and Dustin Axman. Probabilistic Extension of Precision, Recall, and F1 Score for More Thorough Evaluation of Classification Models. In *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, pages 79–91, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.eval4nlp-1.9. URL `https://www.aclweb.org/anthology/2020.eval4nlp-1.9`.

[14] Richard Dinga, Brenda W.J.H. Penninx, Dick J. Veltman, Lianne Schmaal, and Andre F. Marquand. Beyond accuracy: Measures for assessing machine learning models, pitfalls and guidelines. preprint, Neuroscience, August 2019. URL `http://biorxiv.org/lookup/doi/10.1101/743138`.

[15] Robin Wåsjø. Object Recognition and Segmentation of Wounds. 2015. URL `https://www.duo.uio.no/handle/10852/43866`.

[16] Yunyang Xiong, Hyunwoo J. Kim, Bhargav Tangirala, Ronak Mehta, Sterling C. Johnson, and Vikas Singh. On Training Deep 3D CNN Models with Dependent Samples in Neuroimaging. In Albert C. S. Chung, James C. Gee, Paul A. Yushkevich, and Siqi Bao, editors, *Information Processing in Medical Imaging*, Lecture Notes in

Computer Science, pages 99–111, Cham, 2019. Springer International Publishing. ISBN 9783030203511. doi: 10.1007/978-3-030-20351-1_8.

[17] Cyril Goutte and Eric Gaussier. A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Dough Tygar, Moshe Y. Vardi, Gerhard Weikum, David E. Losada, and Juan M. Fernández-Luna, editors, *Advances in Information Retrieval*, volume 3408, pages 345–359. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005. ISBN 978-3-540-25295-5 978-3-540-31865-1. doi: 10.1007/978-3-540-31865-1_25. URL `http://link.springer.com/10.1007/978-3-540-31865-1_25`. Series Title: Lecture Notes in Computer Science.

[18] A. Anjos, L. El Shafey, R. Wallace, M. Günther, C. McCool, and S. Marcel. Bob: a free signal processing and machine learning toolbox for researchers. In *20th ACM Conference on Multimedia Systems (ACMMM), Nara, Japan*, October 2012. URL `https://publications.idiap.ch/downloads/papers/2012/Anjos_Bob_ACMMM12.pdf`.

[19] Bayes' theorem, February 2022. URL `https://en.wikipedia.org/w/index.php?title=Bayes%27_theorem&oldid=1072734317`. Page Version ID: 1072734317.

[20] Isabelle Albert and Jean-Baptiste Denis. Dirichlet and multinomial distributions: properties and uses in Jags. pages 11–12.

[21] Monte Carlo method, February 2022. URL `https://en.wikipedia.org/w/index.php?title=Monte_Carlo_method&oldid=1073500639`. Page Version ID: 1073500639.

[22] Dr Dennis Robert MBBS MMST. Five Confidence Intervals for Proportions That You Should Know About, August 2020. URL `https://towardsdatascience.com/five-confidence-intervals-for-proportions-that-you-should-know-about-7ff5484c024f`.

[23] Binomial proportion confidence interval, February 2022. URL `https://en.wikipedia.org/w/index.php?title=`

Binomial_proportion_confidence_interval&oldid=1070020500. Page Version ID: 1070020500.

[24] Edwin B. Wilson. Probable Inference, the Law of Succession, and Statistical Inference. *Journal of the American Statistical Association*, 22(158):209–212, June 1927. ISSN 0162-1459, 1537-274X. doi: 10.1080/01621459.1927.10502953. URL http://www.tandfonline.com/doi/abs/10.1080/01621459.1927.10502953.

[25] C. J. Clopper and E. S. Pearson. THE USE OF CONFIDENCE OR FIDUCIAL LIMITS ILLUSTRATED IN THE CASE OF THE BINOMIAL. *Biometrika*, 26(4):404–413, 1934. ISSN 0006-3444, 1464-3510. doi: 10.1093/biomet/26.4.404. URL https://academic.oup.com/biomet/article-lookup/doi/10.1093/biomet/26.4.404.

[26] Alan Agresti and Brent A. Coull. Approximate Is Better than "Exact" for Interval Estimation of Binomial Proportions. *The American Statistician*, 52(2):119, May 1998. ISSN 00031305. doi: 10.2307/2685469. URL https://www.jstor.org/stable/2685469?origin=crossref.

[27] Stephanie. Wald CI, February 2022. URL https://www.statisticshowto.com/wald-ci/.

[28] Cross-validation (statistics), February 2022. URL https://en.wikipedia.org/w/index.php?title=Cross-validation_(statistics)&oldid=1073124250. Page Version ID: 1073124250.

[29] Markov chain Monte Carlo, February 2022. URL https://en.wikipedia.org/w/index.php?title=Markov_chain_Monte_Carlo&oldid=1073751895. Page Version ID: 1073751895.

[30] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, April 1970. ISSN 1464-3510, 0006-3444. doi: 10.1093/biomet/57.1.97. URL https://academic.oup.com/biomet/article/57/1/97/284580.

[31] Metropolis–Hastings algorithm, November 2021. URL https://en.wikipedia.org/w/index.php?title=Metropolis%E2%80%93Hastings_algorithm&oldid=1056490069. Page Version ID: 1056490069.