HMM-SMC Project :

PAC-Bayesian AUC classification and scoring

Hamdi BEL HADJ HASSINE, *Alexandre* PONCIN, *Georges* SARR

I Introduction

Binary classification is a common problem in statistics with applications in many fields in research and in industry. While logistic regression is generally considered the standard method for this problem, alternative models have been proposed by the literature to achieve better results on unbalanced data or to optimize different target criteria like AUC. In this project we study the PAC-Bayesian approach proposed by James Ridgway, Pierre Alquier, Nicolas Chopin and Feng Liang. This approach targets the bipartite scoring problem : Given a dataset $(X_i, Y_i)_{i\in 1..n}$ with binary labels $Y_i \in \{-1, 1\}$, compute a score $s(X_i)$ for every observation X_i such that when ranking two samples $(X^+, 1)$ and $(X^-, -1)$ with respectively positive and negative labels, the probability that $s(X^+) > s(X^-)$ is maximized. In other words, we would like to compute scores such that positive observations $(Y = 1)$ have higher scores than negative observations $(Y = -1)$. Once we have computed the scores of the observations, binary classification is reduced to the choice of a threshold t such that observations having $s(X_i) \leq t$ are classified as negative and observations having $s(X_i) > t$ are classified as positive. This choice of the threshold is equivalent to choosing a point on the ROC curve and gives finer control of the false negative and false positive rates, while under mild assumptions, the previously stated maximization problem is equivalent to maximazing the AUC (Area Under Curve) metric [1].

II PAC-Bayesian Scoring Problem Definition

II.1 Linear score function

Our goal for this project is to apply SMC sampling to solve the PAC-Bayesian scoring problem. For this, we first define the AUC risk function that we would like to minimize :

$$R(s) = \mathbb{P}_{(X,Y),(X',Y')\sim P}\left[\left(s(X) - s\left(X'\right)\right)\left(Y - Y'\right) < 0\right]$$

and its empirical counterpart :

$$R_n(s) = \frac{1}{n(n-1)} \sum_{i\neq j} \mathbb{1}\left[\left(s\left(X_i\right) - s\left(X_j\right)\right)\left(Y_i - Y_j\right) < 0\right]$$

In this section we study linear score functions with parameter $\theta = (\theta_1, ..., \theta_d)^T \in \mathbb{R}^d$:

$$s_\theta(X) = \langle X, \theta\rangle$$

The problem consists therefore in computing a coefficient vector θ that minimizes $R(s_\theta)$ (which will be denoted $R(\theta)$). To do this we define a prior distribution $\pi_\xi(\theta)$ from which we derive the following posterior distribution using the available data :

$$\pi_{\xi,\gamma}(\theta \mid \mathcal{D}) := \frac{\pi_\xi(\theta)\exp\left\{-\lambda R_n(\theta)\right\}}{Z_{\xi,\lambda}(\mathcal{D})}, \quad Z_{\xi,\lambda}(\mathcal{D}) = \int_{R^d} \pi_\xi(\tilde{\theta})\exp\left\{-\lambda R_n(\tilde{\theta})\right\} \mathrm{d}\tilde{\theta}$$

Here the prior distribution is usually chosen to be a multivariate normal which is shown in the paper to have optimal excess risk rate. It is also possible to use a spike and slab prior to encourage sparsity.of the solution.

The posterior distribution is composed by the prior divided by a normalizing constant, and an exponential term that penalizes $R(\theta)$. Therefore intuitively, values that have higher posterior density

(more likely to be sampled) are values that have lower risk. The tempering parameter λ determines how much the risk is penalized : low values will lead to higher risk θ values to be more likely, while high values would make the posterior distribution more different from the prior distribution and therefore harder to sample from. This trade-off problem can be solved by sampling from the posterior distribution using SMC tempering, an algorithm which adaptively increases λ while sampling from a sequence of distributions interpolating between the prior and the posterior.

As a result, this method allows us to compute not only the parameter vector θ but also the full distribution of each one of its components $\theta_1, ..., \theta_d$ which can be useful to diagnose the model and evaluate the variance of θ.

II.2 Nonlinear Extension

The PAC-Bayesian AUC classification can be extended to non-linear scoring function with the pseudo-posterior

$$\pi_{\xi,\gamma}(\mathrm{d}s|\mathcal{D}) \propto \pi_{\xi}(\mathrm{d}s) \exp\left\{-\frac{\gamma}{n^+n^-} \sum_{i\in\mathcal{D}^+, j\in\mathcal{D}^-} \mathbb{1}\left\{s\left(X_i\right) - s\left(X_j\right) > 0\right\}\right\}$$

where $\pi_{\xi}(\mathrm{d}s)$ is a prior probability measure over an infinite functional set. Assume that $\pi_{\xi}(\mathrm{d}s)$ follows a gaussian process associated to a kernel $k_{\xi}(x, x')$. Let $s_i := s(X_i)$ and $s_{1:n} := (s_1 \dots s_n)$, we find in the literature the marginal posterior formula

$$\pi_{\xi,\gamma}(\mathrm{d}s|\mathcal{D}) \propto \mathcal{N}_d(s_{1:n}, 0, K_{\xi}) \exp\left\{-\frac{\gamma}{n^+n^-} \sum_{i\in\mathcal{D}^+, j\in\mathcal{D}^-} \mathbb{1}\left\{s_i - s_j > 0\right\}\right\}$$

with $K_{\xi} := \left(k_{\xi}(X_i, X_j)\right)_{1\leq i,j\leq n}$.

With this structure for the marginal posterior, the same approach as in the linear case can be applied with $s_{1:n}$ as parameter of dimension n. Then the score function estimate is extended to any points using a gaussian process regression.

III SMC Tempering

III.1 Idea

The goal is to sample from the posterior distribution $\pi_{\xi,\gamma}$ given the dataset $\mathcal{D}$. Tempering is a way to do so. The basic idea is to go from a prior distribution π_{ξ} which is easier to sample from and then sequentially approximate intermediary distributions $\pi_{\xi,\gamma_1}, ..., \pi_{\xi,\gamma_T}$ such that $\gamma_1 < ... < \gamma_T$, so that the final distribution π_{ξ,γ_T} is an approximation of the posterior distribution.

The tempering algorithm approximates $\pi_{\xi,\gamma_{t-1}}$ by yielding N particles θ^i_{t-1} and their corresponding weights $w(\theta^i_{t-1})$. In order to move from $\pi_{\xi,\gamma_{t-1}}$ to π_{ξ,γ_t}, importance sampling is performed. In other words the weight of each particle θ gets adjusted proportionately to

$$\frac{\pi_{\xi,\gamma_t}(\theta)}{\pi_{\xi,\gamma_{t-1}}(\theta)} \propto \exp\left[-\left(\gamma_t - \gamma_{t-1}\right) R_n(\theta)\right] = w_t(\theta)$$

in order to, loosely speaking, "adapt" the particles to π_{ξ,γ_t}. More formally, the distribution π_{ξ,γ_t} is chosen such that the Effective Sample Size (ESS) is proportional to the number N of particles : that

is $ESS = \tau N$, where $\tau \in (0,1)$ is a fixed hyperparameter, doing so ensures the particles weights not to degenerate too much. A resampling step (with replacement) is carried out afterwards to keep, with higher probability, particles with large weights. Finally K MCMC steps are performed to get a variety of particles.

Tempering can be used to choose the exponent γ among the γ_t's. In fact when the algorithm produces particles at each time t, their weighted mean θ_t can be computed. We can then use θ_t to compute the corresponding AUC and then choose the parameter γ_t for which the corresponding weighted mean θ_t maximises the AUC. We use this technique to monitor the γ_t values and evaluate which range of values provide better results. However we still need to choose γ_T beforehand, and to do so we use cross-validation as suggested in the paper.

In the upcoming sections we will discuss how to choose the other hyperparameters.

III.2 Algorithm

Algorithm 1 Tempering SMC

Input N (number of particles), $\tau \in (0,1)$ (ESS threshold), $\kappa > 0$ (random walk tuning parameter)
Init. Sample $\theta_0^i \sim \pi_\xi(\theta)$ for $i = 1$ to N, set $t \leftarrow 1, \gamma_0 = 0, Z_0 = 1$.
Loop

a. Solve in γ_t the equation

$$\frac{\left\{\sum_{i=1}^N w_t\left(\theta_{t-1}^i\right)\right\}^2}{\sum_{i=1}^N w_t\left(\theta_{t-1}^i\right)^2} = \tau N, \quad w_t(\theta) = \exp\left[-\left(\gamma_t - \gamma_{t-1}\right) R_n(\theta)\right]$$

using bisection search. If $\gamma_t \geq \gamma_T$, set $Z_T = Z_{t-1} \times \left\{\frac{1}{N}\sum_{i=1}^N w_t\left(\theta_{t-1}^i\right)\right\}$, and stop.

b. Resample : for $i = 1$ to N, draw A_t^i in $1, \ldots, N$ so that

$$\mathbb{P}\left(A_t^i = j\right) = w_t\left(\theta_{t-1}^j\right) / \sum_{k=1}^N w_t\left(\theta_{t-1}^k\right)$$

c. Sample $\theta_t^i \sim M_t\left(\theta_{t-1}^{A_t^i}, \, \mathrm{d}\theta\right)$ for $i = 1$ to N where M_t is a MCMC kernel that leaves invariant π_t

d. Set $Z_t = Z_{t-1} \times \left\{\frac{1}{N}\sum_{i=1}^N w_t\left(\theta_{t-1}^i\right)\right\}$.

The package used to experiment this algorithm sets by default the random walk tuning parameter κ to some constant that comes frequently in the literature which allegedly works often well in practice. We did not change that value.

IV Methodology

In order to evaluate the proposed SMC-based PAC-Bayesian classification algorithm, we evaluate it on multiple datasets and compare the results with logistic regression, which is a standard benchmark for binary classification.

In this project we used the Particles package to perform Tempering SMC sampling from the posterior distribution. First we define the prior distribution as a multivariate Gaussian with scale hyperparameter ξ. We then implemented the risk function as defined in the paper and we took care to optimize it using the Numba package (achieving a 15x speedup compared to a standard Numpy implementation). Then we defined the particle filter which samples from the posterior distribution using waste-free Tempering SMC [2]. The output of each run is a set of $(K+1)N$ θ vectors, which we can use to plot the posterior distribution for each coefficient θ_i. In order to make predictions, we chose to use the modal value for each θ_i distribution for two reasons : the first is that taking the mean or median would yield poor prediction results when the posterior is bimodal, and the second is that if we assume that the prior's contribution to the posterior is negligible (for λ high enough), then the mode is the θ value that minimizes the risk. Therefore we define θ_M as the component-wise modal value of the sampled θ vectors, then we use it to make predictions $(\langle \theta_M, X_i \rangle)_i$ and calculating the AUC.

It remains to choose the hyperparameter values for the sampling which are K (MCMC steps), N (number of particles), $\lambda_{\max}$ (maximum tempering parameter), ESSrmin (resampling threshold) and ξ (scale of the prior Gaussian). To do this we divided the dataset into training (75%) and test (25%) and we used 5-fold cross-validation on the training data to optimize the hyperparameter values sequentially : We fix 4 hyperparameters and we do cross-validation on different values of the fifth hyperparameter, then we choose the value that yields the highest mean AUC over the 5 validation folds. We use this method to optimize the hyperparameters one by one until convergence of the AUC values. Once we have selected the best-performing hyperparameter values, we use them to predict the test data and calculate the AUC, which we compare with logistic regression.

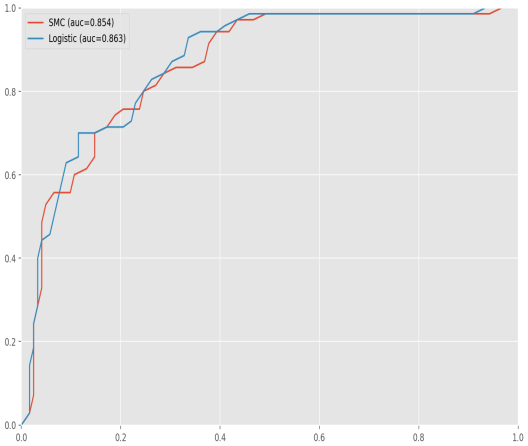
V Results

Dataset	Covariates	Balance	AUC	AUC std.	Logit AUC
Pima	7	34%	0.8598	0.0028	0.863
Sonar	60	47%	0.6967	0.0352	0.711
EEG	14	45%	0.651	0.004	0.625

TABLE 1 – Comparison of AUC for the three studied datasets

(a) (b) (c)

FIGURE 1 – Marginal posterior of the first three coefficients for Pima dataset



(a)

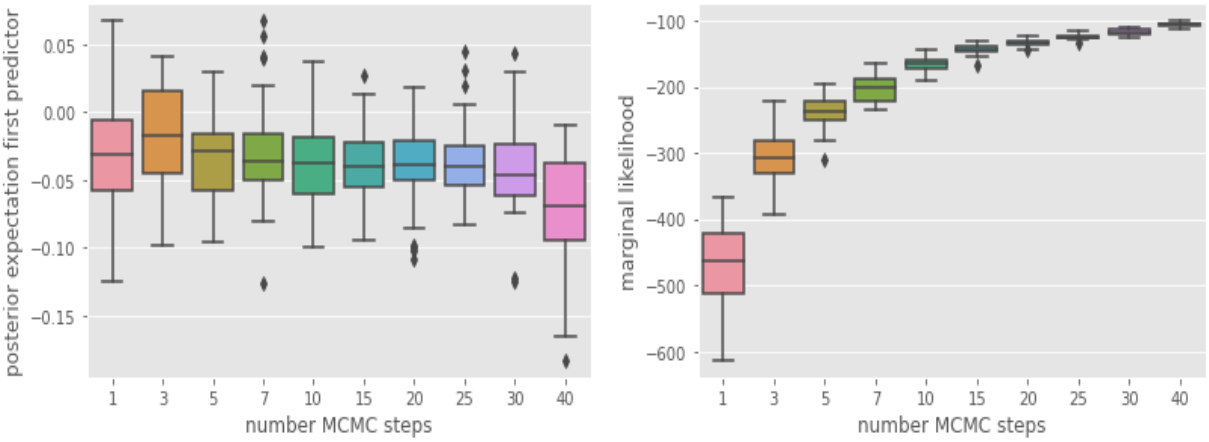
FIGURE 2 – ROC Curve on Pima dataset

We observe that the PAC-Bayesian classifier obtains results similar to logistic regression. For EEG we get an AUC 4% better than logistic regression, while for Sonar it performs 2% worse.

We also implemented the non-linear classifier but it didn't perform well in our tests. Most of the time, the sampler returns an error during the resampling calibration step which indicates that the covariance matrix might be not numerically stable.

VI SMC Sampler diagnostics

In order to check how well tempering behaves, we plotted different metrics against the number of MCMC steps.



(a) (b)

FIGURE 3 – Posterior expectation of the first predictor and marginal likelihood using 50 tempering runs for each number of MCMC steps on the Sonar dataset with marginal priors $\mathcal{N}(0, 0.1^2)$ and $N = 1000$.

On the Sonar dataset, the posterior expectation average estimate for the first predictor (coefficient) seems quite stable for relatively small values of K, and looks like setting high values of values give poor estimates of the posterior expectation. However the marginal likelihood estimation seems to behave better when the number K of MCMC steps grows which is indicated by smaller box plot size (so smaller variability) as K gets larger.

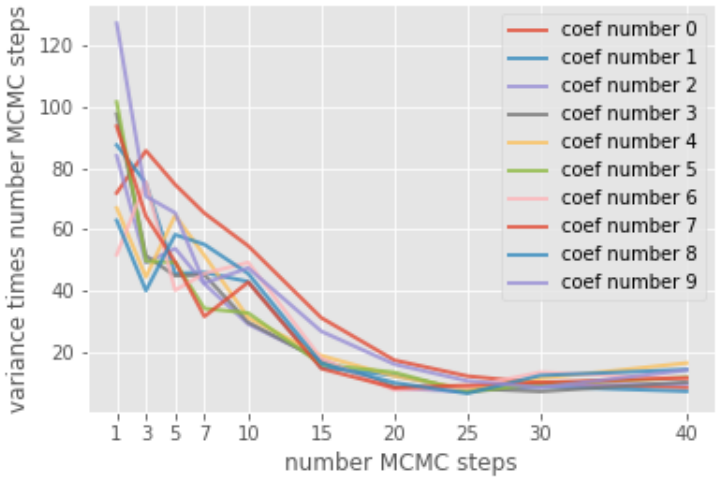


FIGURE 4 – Adjusted variance of the posterior expectation for the 10 first coefficients (coordinates of the parameter θ) against the number of MCMC steps

For each coefficient, the adjusted variance is equal to K times the ratio of the variance of its posterior expectation estimates and the average estimate of its posterior variance. It accounts for the CPU time through K and for the sampler preciseness through the ratio, such that the larger it gets, the worse the sampling seems. On the Sonar dataset, it appears that setting K large enough is the best compromise, for example $K = 15$.

VII Conclusion

We implemented a PAC-Bayesian AUC classification and scoring algorithm that uses SMC tempering to sample from a parameter distribution that minimizes the empirical AUC risk.

We tested the classifier on 3 different datasets and obtained varying results depending on the dataset. We can conclude that PAC-Bayesian classification provides an alternative to logistic regression that can offer better results on some datasets.

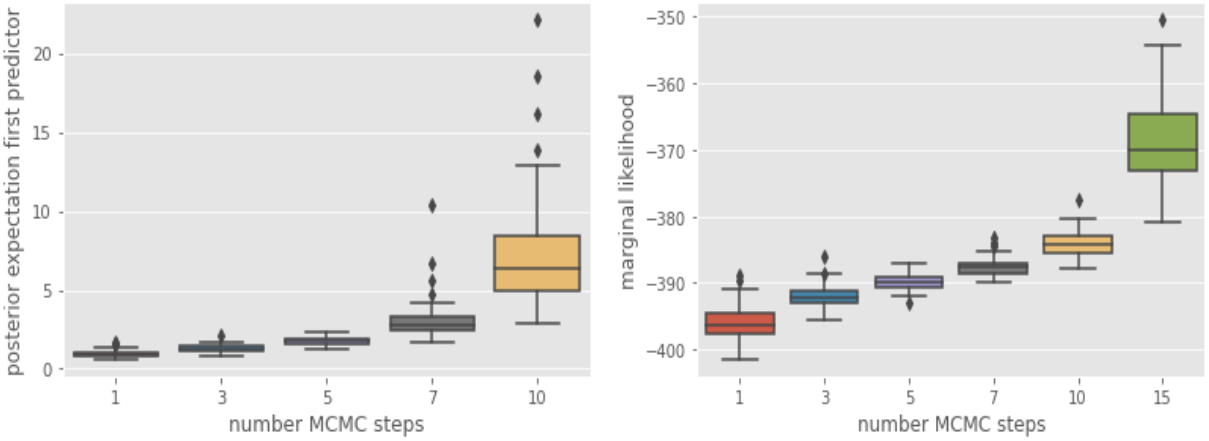
References

[1] C. Cortes and M. Mohri, "Auc optimization vs. error rate minimization," vol. 16, 2004. [Online]. Available : https ://proceedings.neurips.cc/paper/2003/file/6ef80bb237adf4b6f77d0700e1255907-Paper.pdf

[2] H.-D. Dau and N. Chopin, "Waste-free sequential monte carlo," 2021.

Appendices

A Diagnostics of the sampler on PIMA dataset



(a) (b)

FIGURE 5 – Posterior expectation of the first predictor and marginal likelihood using 50 tempering runs for each number of MCMC steps on the PIMA dataset

On the PIMA dataset, the posterior expectation for the first predictor seems to vary more when the number of MCMC steps increases. That is quite surprising, we would expect the inverse to happen since the larger the number of MCMC steps the more diverse the particles are, hence more stable estimates of the expectation. The same goes for the marginal likelihood. One possible explanation is the small size of the PIMA dataset.

B Gaussian regression

We want to make predictions $y_2 = s(X_2)$ for a test set X_2 based on our Gaussian process prior and n_1 previously observed points $(X_1, y_1 = s_{1:n})$ from a training set. This can be done with the help of the posterior distribution $p(y_2|X_1, y_1, X_2)$. They come from the same multivariate gaussian distribution :

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}\right)$$

In the case of our gaussian process, y follows distribution $\mathcal{N}(0, k_\xi(X, X))$ for any (X, y) hence

$$\begin{aligned} \mu_1 &= 0 \\ \mu_2 &= 0 \\ \Sigma_{ij} &= k_\xi(X_i, X_j) \end{aligned}$$

One deduce the conditional distribution $p(y_2|X_1, y_1, X_2) = \mathcal{N}(\mu_{2|1}, \Sigma_{2|1})$ with

$$\mu_{2|1} = \mu_2 + \Sigma_{21}\Sigma_{11}^{-1}(y_1 - \mu_1) = \Sigma_{21}\Sigma_{11}^{-1}y_1$$
$$\Sigma_{2|1} = \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}$$

It is then possible to predict y_2 by using $\mu_{2|1}$.