Available online at www.eccomasproceedia.org
Eccomas Proceedia UNCECOMP (2017) 339-349

ECCOMAS
Proceedia

UNCECOMP 2017
2nd ECCOMAS Thematic Conference on
Uncertainty Quantification in Computational Sciences and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou (eds.)
Rhodes Island, Greece, 15–17 June 2017

BAYESIAN UPDATING FOR PROBABILISTIC CLASSIFICATION USING RELIABILITY METHODS

P.G.Byrnes and F.A. DiazDelaO[1]

[1]Institute for Risk and Uncertainty, University of Liverpool,
Liverpool, UK
e-mail: [1]{paul.byrnes,f.a.diazdelao}@liverpool.ac.uk

Keywords: Bayesian Updating, Reliability Analysis, Subset Simulation, Machine Learning, Gaussian Process Classification

Abstract. *Probabilistic classification requires the computation of the posterior probability distribution of a class given a data observation. In order to generate posterior samples, an analogy has recently been established between the Bayesian updating problem and the engineering reliability problem which allows reliability methods to be applied to the former. The modification of the BUS (Bayesian Updating with Structural Reliability Methods) formulation is based on the conventional rejection principle and suggests the application of Subset Simulation (SuS) from reliability engineering to sample from posterior distributions. Under the original BUS framework a likelihood multiplier is required to be calculated before the implementation of SuS. A recently proposed algorithm learns the likelihood multiplier automatically. This research proposes the utilization of BUS for Gaussian Process classification. The above framework is illustrated using a benchmark Machine Learning dataset from an engineering application.*

Peer-review under responsibility of the organizing committee of UNCECOMP 2017.
doi: 10.7712/120217.5374.16922

1 INTRODUCTION

Classification is a branch of supervised Machine Learning which allocates test instances to a class based on a training dataset. Let C_k be the k^{th} class from a family $\{C_1, ..., C_k\}$ and D be a data set. Define the posterior probability distribution of interest as $P(C_k|D)$. Various techniques exist for sampling from and approximating a posterior distribution. Popular methods include Markov Chain Monte Carlo (MCMC) [1], Laplace Approximation [2], Expectation Propagation (EP) [3] and Rejection sampling [4]. Rejection sampling consists of drawing a random sample from the prior distribution, computing the likelihood and accepting the sample proportional to the likelihood. However, the disadvantage of this method is that the acceptance rate can be extremely low. This opens the possibility as viewing posterior sampling as rare event simulation.

An analogy has recently been established between the Bayesian updating problem and the engineering reliability problem. The formulation, called BUS (Bayesian Updating with Structural reliability methods) [5] is based on the conventional rejection principle and allows reliability methods to sample from posterior distributions. Through the realisation that the probability of acceptance in rejection sampling is equivalent to the probability of failure in a reliability problem, it is concluded that reliability methods may be applied to Bayesian updating problems. Subset Simulation (SuS) [6] is an advanced Monte Carlo technique used in reliability engineering which estimates the probability of a rare event. SuS expresses the (rare) failure event F as contained in a nested sequence of more frequent events. This enables the algorithm to calculate the probability of failure. SuS has thus been shown to be a robust technique which is suitable for Bayesian computations.

One problem which stems from the original BUS framework is that SuS is dependent on the choice of a constant referred to as the likelihood multiplier. Some suggestions regarding the calculation of the multiplier have been given [5]. However, correctly choosing the value of this multiplier has in general remained an open question. A revised BUS formulation [7] allows SuS to be implemented to sample from the posterior distribution whilst learning the multiplier automatically. The objective of this paper is to apply the modified BUS framework to sample from a posterior distribution in a binary classification setting. The classification framework implemented depends on a Gaussian Process (GP) model. The data set used is the widely applied 'Ionosphere' data set [8].

The organization of this paper is as follows. SuS along with the BUS methodologies are introduced in section 2. Section 3 presents the GP classifier along with experimental results. Section 4 contains conclusions and comments on future work.

2 METHODOLOGY

Given that the BUS methodology identifies a relationship between the Bayesian updating problem and the engineering reliability problem, the following section will first provide a brief overview of reliability analysis before introducing SuS and the BUS framework.

2.1 Reliability Analysis

The behaviour of a system may be represented by a response variable Y which is dependent on input variables $x = (x_1, ..., x_d)$ such that

$$Y = g(x_1, ..., x_d) \tag{1}$$

where d represents the dimension of the problem and $g(x)$ the performance function. In this setting, the failure event F occurs when the output of Y exceeds a critical threshold b. This may be expressed as

$$F = \{x : g(x) > b\} \tag{2}$$

Let $\pi(x)$ denote the joint PDF for x. The engineering reliability problem is to compute the probability of failure $P(F)$, given by

$$P(F) = P(x \in F) = \int_F \pi(x)dx \tag{3}$$

2.2 Subset Simulation

SuS is a widely used simulation technique in reliability analysis for simulating rare events and estimating failure probabilities. By expressing the rare event F as an intersection of nested events, $F = F_m \subset F_{m-1} \subset ... \subset F_1$, where F_m is rare event whilst F_1 may be viewed as a rather frequent event, the algorithm estimates the value of $P(F)$. Given this decomposition, it is easily shown that

$$P(F) = P(F_m|F_{m-1}) * P(F_{m-1}|F_{m-2}) * ... * P(F_2|F1) * P(F_1) \tag{4}$$

By an appropriate selection, the probabilities $P(F_1)$ and $P(F_{i+1}|F_i)$ can be estimated by direct Monte Carlo. Hence, the original rare event problem is broken down into a series of intermediate subproblems which define a series of intermediate thresholds. SuS seeks to estimate the complementary cumulative distribution function (CCDF) of a response quantity, that is $P(Y > b)$. The CCDF is viewed simply as the tail of the distribution of an 'exceedance' area. This CCDF can be used directly for estimating the failure probability that the response exceeds a specified threshold b. Let $p_0 \in [0, 1]$ be the level probability which in essence governs how many intermediate failure thresholds are required to reach the failure domain F. In the reliability literature, a sensible choice is $p_0 \in [0.1, 0.3]$ [9]. Let $N \in \mathbb{N}$ be the total number of generated samples. While $n_s = p_0 N \in \mathbb{N}$ governs the required number of accepted samples at each level. SuS requires the parameters p_0 and N be determined before beginning the simulation.

Beginning at $level_0$, the algorithm probes the input space generating N independent and identically distributed (i.i.d) samples by direct Monte Carlo methods. Based on the values of Y computed by the performance function, the first intermediate failure threshold b_1 is calculated. The n_s samples which exceed b_1 from $level_0$ are thus stored as seeds for generating additional samples conditional on $F_1 = \{Y > b_1\}$ at $level_1$. For $level_1$ a MCMC algorithm is utilized to populate F_1. Similar to $level_0$ the samples which exceed b_2 are used as seeds for the next level. The generation of intermediate levels is continued in the same manner until F is populated with the pre-defined number of samples where the probability of failure is approximated by

$$P(F) \approx p^l \frac{n_F^l}{N} \tag{5}$$

where l represents the termination level and n_F the number of failure samples at that level. For more details on the implementation of SuS, refer to [6].

2.3 BUS

The BUS formulation builds a relationship between the Bayesian updating problem and the engineering reliability problem, thereby allowing reliability methods (in this case SuS) to be applied to the former. Consider a continuous random variable X. Let $L(x)$ denote the likelihood function, $q(x)$ denote the prior PDF , P_D denote the normalizing constant $\int L(x)q(x)dx^{-1}$ and $P(x)$ denote the posterior PDF. Thus,

$$P(x) = P_D^{-1} L(x)q(x) \tag{6}$$

The Rejection sampling scheme generates a sample from $P(x)$ as follows:

1. Generate U uniformly distributed on [0,1] and x with the prior PDF $q(x)$.
2. If $U < cL(x)$, return x as the sample. Otherwise repeat step 1.

where c is a constant such that the rejection principle inequality $(cL(x) \leq 1)$ holds. In the context of the Bayesian updating problem, let the driving variable of the engineering reliability problem be defined as follows:

$$Y = cL(x) - u \tag{7}$$

where u is a standard uniform random variable in [0,1]. The corresponding failure event

$$F = \{Y > 0\} \tag{8}$$

can be sampled from using SuS. It is a well-known fact that in the calculation of c by the rejection principle, the largest admissible value of c is given by

$$c_{\max} = \frac{1}{\max_x L(x)} \tag{9}$$

In the case of the value being greater than $c_{\max}$, the posterior generated samples contain bias. A smaller value will still produce the correct samples but will be less efficient [7].

2.4 Modified Bus Framework

A modification of the original BUS formulation has been proposed [7] which isolates the effect of the multiplier on the simulation. The modification involves a reexpression of the failure event such that

$$F = \left\{ \ln\left[\frac{L(x)}{u}\right] > -\ln c \right\} \tag{10}$$

The above formulation results in the driving variable Y in SuS being defined as

$$Y = \ln\left[\frac{L(x)}{u}\right] \tag{11}$$

Adjustments to how F and Y are defined allow the target failure event to be expressed as $F = \{Y > b\}$, where $b = -\ln c$. It is evident that as Y is no longer dependent on c it is not necessary to choose the value of c before SuS runs. The multiplier only affects the target threshold level b beyond which the samples can be collected as posterior samples.Under the new framework the distribution of the samples conditional on $\{Y > b\}$ will remain unchanged for a sufficiently large b. The minimum value of b beyond which the distribution of the samples will settle at the posterior PDF can be shown to be

$$b_{\min} = -\ln c_{\max} = \ln[\max_x L(x)] \tag{12}$$

2.4.1 Identification of Minimum Threshold Level

Similar to $c_{\max}$ the value of $b_{\min}$ is unknown.. The behaviour of $P(Y > b)$ as b varies can be examined during a SuS run to determine whether the threshold value of a particular level has passed $b_{\min}$. Regarding the CCDF, when b is at the left tail of distribution then $P(Y > b) \approx 1$. The value of $P(Y > b)$ typically decreases with b equal to P_D at $b = b_{\min}$. When $b > b_{\min}$ it can be shown that $P(Y > b) = e^{-b} P_D$ [7] where $P_D = \int q(x)L(x)dx$.

It can be expected that as b increases from the left tail and to a value greater than $b_{\min}$, the CCDF of Y typically changes from a decreasing function to an exponentially decaying function. Correspondingly, the function $\ln P(Y > b)$ changes from a slowly decreasing function to a straight line with a slope of -1. Additionally, consider the following:

$$V(b) = b + lnP(Y > b) \tag{13}$$

This function can be used for computing the log-evidence in $\ln P_D$ as it can be observed that

$$V(b) = \ln P_D \qquad b > b_{\min} \tag{14}$$

When b is at the left tail of the CCDF, $\ln P(Y > b) \approx 0$ and so $V(b) \approx b$ increases linearly with b. In other words as b increases from the left tail of the CCDF of Y the function $V(b)$ increases linearly, going through a transition until it settles at $\ln P_D$ after $b > b_{\min}$. Figure 1 contains a graphical representation of how $\ln P(Y > b)$ and $V(b)$ change when SuS has surpassed $b_{\min}$. Both above quantities b can only be estimated on a sample basis.

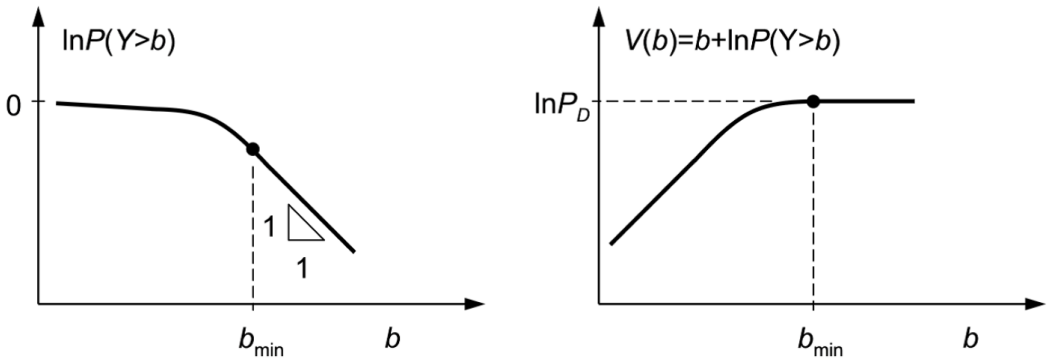

Figure 1: Theoretical characteristic trends of $\ln P(Y > b)$ and $V(b)$.

2.4.2 Automatic Stopping Condition

On the basis of the above characteristic trends an automatic stopping condition can be implemented once the algorithm detects that the transition has occurred. Consider an inadmissible level m such that $b_m < b_{min}$. There exists a constant e^{-b_m} and a monotone decreasing sequence a_m

$$lim_{m\to\infty} \quad a_m = 0 \tag{15}$$

where a_m is the prior probability of the inadmissible set $B_m = \{x : e^{-b_m} L(x) > 1\}$. Through the expression of the marginal distribution of the target variable as

$$p(x|F_m) \propto \begin{cases} q(x), & \text{if } x \in B_m \\ e^{-b_m} q(x) L(x), & \text{if } x \in B_m^c \end{cases}$$

$$P_{F_m} = P_x(B_m) + e^{-b_m} P_D P_{x|D}(B_m^c) \tag{16}$$

it is observed that for $x \in B_m$ the marginal is proportional to the prior distribution. Given that the prior distribution is a probability measure it satisfies the monotonicity property. Therefore it follows that a_m is a monotone decreasing sequence of value which will converge to zero. Through the expression of the prior probability as

$$a_m = P_x(B_m) = P_x(L(x) > e^{b_m}) \tag{17}$$

the stopping condition takes the form of a reliability problem that in turn may also be solved by SuS.

3 GAUSSIAN PROCESS BINARY CLASSIFICATION

Due to the non-parametric nature of Gaussian processes, their use for Machine Learning applications has grown greatly in recent years. By focussing on processes which are Gaussian, it turns out that the computations required for inference and Machine Learning become relatively easy [10] in comparison to other methods e.g Neural Networks. A GP model is defined by its mean and covariance functions respectively such that

$$GP \sim \mathcal{N}(m(x), k(x, x')) \tag{18}$$

Bayesian inference in a GP classification model is performed about the latent function f having observed data $D = \{(y_i, \mathbf{x}_i)|i, ..., n\}$. Let $f = [f_1, ...f_m]^T$ represent the values of the latent function, $d = [d_1, ..., d_m]^T$ the model inputs and $y = [y_1, ..., y_m]^T$ the class labels where $y \in \{-1, 1\}$. Given the latent function, the class labels are independent Bernoulli random variables. It can be shown [11] that the likelihood can be factorized as

$$p(y|f) = \prod_{i=1}^{m} p(y_i|f_i) = \prod_{i=1}^{m} \Phi(y_i f_i) \tag{19}$$

where Φ represents the CDF of the standard Gaussian distribution. The latent function is given a GP prior which implies that any finite subset of latent variables has a multivariate Gaussian

distribution [10]. Often in a binary setting since neither of the class labels is more probable the mean of the prior over f is set to zero. A covariance function of the form $k(x, x'|\theta)$, where θ represents the functions hyperparameters is combined with the zero mean function to define the GP. Through the application of Bayes rule the posterior distribution over the latent function f for given hyperparameters θ is expressed as

$$p(f|D,\theta) = \frac{p(y|f)p(f|X,\theta)}{p(D|\theta)} = \frac{\mathcal{N}(f|0,K)}{p(D|\theta)} \prod_{i=1}^{m} \Phi(y_i f_i) \tag{20}$$

In order to predict y^* from d^* the distribution of the latent function may be computed by marginalising as follows

$$p(f_*|D,\theta,d) = \int p(f_*|f,X,\theta,d)p(f|D,\theta)df \tag{21}$$

which in turn allows the predictive distribution to be obtained by taking the expectation of the marginal

$$p(y_*|D,\theta,d_*) = \int p(y_*|f_*)p(f_*|D,\theta,d_*)df_* \tag{22}$$

In this work, the computation of the above posterior is carried out through the modified BUS framework.

4 EXPERIMENTAL RESULTS

4.1 Ionosphere Data Set

The Ionosphere data set [8] consists of 351 radar observations in 34 dimensions. The training set consists of 200 instances and the test set 151. In a binary classification setting 'Good' radar returns are those showing evidence of some type of structure in the ionosphere and are given the class label $y = 1$. On the other hand 'Bad' returns are those which do not show evidence of a structure type and are given the class label $y = -1$. The inputs are standardized to a zero mean and unit variance.

Expectation Propagation is chosen as a benchmark comparison for this study. Similar to previous benchmark studies [11,12] the covariance function chosen is the squared exponential

$$k(x, x'|\theta) = \sigma^2 exp\left(\frac{||x - x'||^2}{-2l^2}\right) \tag{23}$$

where $\theta = [\sigma, l]$, σ^2 refers to the signal variance and l is the characteristic length scale. The likelihood function used is the CDF of the standard Gaussian distribution or the 'probit' function of the form

$$\Phi(x) = \int_{-\infty}^{x} \mathcal{N}(x|0,1)dx \tag{24}$$

4.2 Results

The parameters for SuS were set to $N = 5,000$ and $p_0 = 0.2$ respectively. A one dimensional proposal PDF was chosen to be uniform in [0,1]. Figure 2 shows the resulting log-CCDF and

log-evidence respectively. The general nature of the curves follows the characteristic trends predicted in Figure 1.

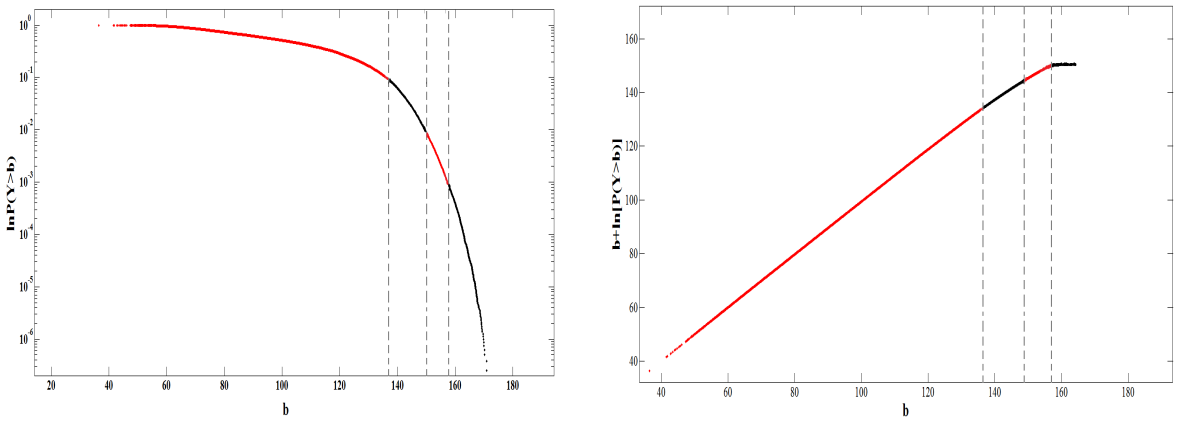

Figure 2: Log-evidence of characteristic trends for SuS sampling. Both $lnP(Y > b)$ (left) and $V(b) = b + lnP(Y > b)$ (right) exhibit behaviour which coincides with the theory. The vertical lines represent the threshold value at each intermediate level.

For the automatic stopping condition a tolerance of $a_m = 10^{-6}$ was set as the threshold for the probability of inadmissibility in Eq.(17) . The evolution of the threshold and the values of the probability of inadmissibility are presented in Table 1. The probability of inadmissibility has converged to a value smaller than the set tolerance at $level_3$. Thus, the samples generated at this level may be assumed to be drawn form the desired posterior distribution.

$Level$	b_m	c_m	a_m
0			
1	138.5642	6.6425e-61	0.9
2	151.6193	1.4209e-66	0.0015
3	159.2372	6.9849e-70	4.89e-07

Table 1: Evolution of the threshold and the probability of inadmissibility

Having computed the posterior distribution over the latent function, the samples generated at $level_3$ are used to produce predictive latent function values f^*. An investigation on a regular 21x21 grid of values for the log hyper-parameters was carried out whereby for each value of θ on the grid the approximate log marginal likelihood is calculated. Figure 3 shows a contour plot of the approximate log marginal likelihood for both BUS and EP.

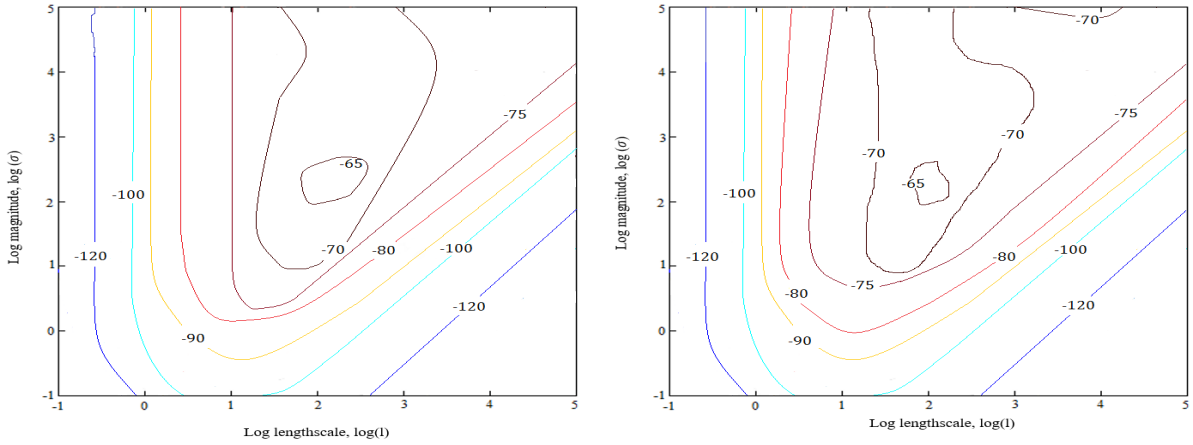

Figure 3: Log marginal likelihoods produced by the modified BUS (left) and Expectation Propagation (EP) (right).

Classification performance on the test data set in terms of predictive probabilities is calculated through the scaled information score, given by

$$I = H + \frac{1}{2n}\sum_{i=1}^{n}(1+y_i)log_2(p_i) + (1-y_i)log_2(1-p_i) \tag{25}$$

where n is the number of test observations. The value of I may range from 0 to 1 where 1 bit indicates perfect prediction and 0 bits random guessing. The information score expresses the difference between the baseline entropy H of the training set labels and the average negative log probabilities. The entropy for the training set labels is given by

$$H = -\sum_{y=+1,-1}\frac{n^y_{test}}{n_{test}}log_2\frac{n^y_{train}}{n_{train}} \leq 1 \tag{26}$$

where n^y_{test} and n^y_{train} denote the number of observations in the test and training data sets repetitively with the given target class label. In this case $y = 1$ and $H = 0.9908$. Had the classes, training and test sets been perfectly balanced H would equal 1. Figure 4 shows a contour plot for BUS and EP where the maximum I values produced were 0.6123 and 0.656 respectively.

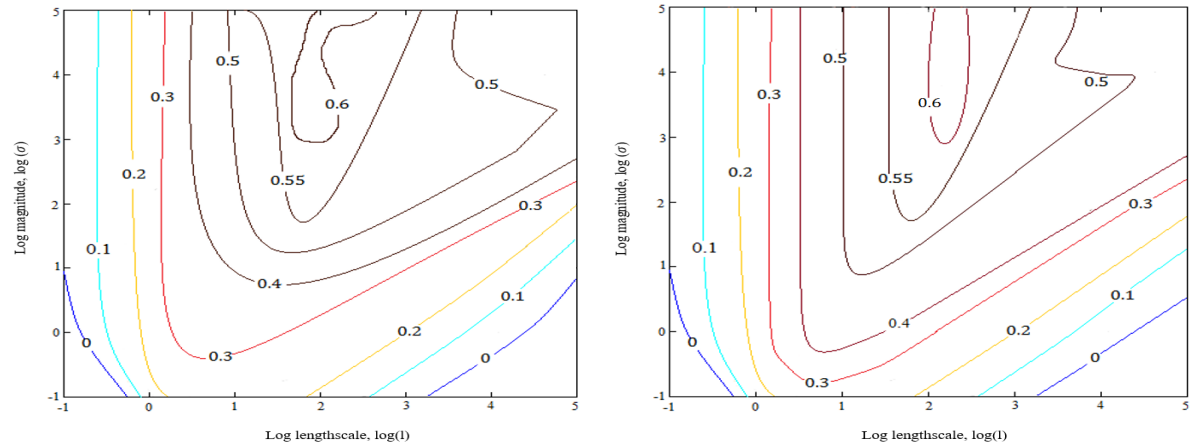

Figure 4: Test information score produced by the modified BUS (left) and Expectation Propagation (EP) (right).

The misclassification rate for 35 independent runs of the modified BUS algorithm using the optimal hyperparameter values from the log marginal likelihood is presented in figure 5. The error threshold used for each test point was 0.5. The average error rate produced was 10.77%. This figure follows closely with other studies [11].

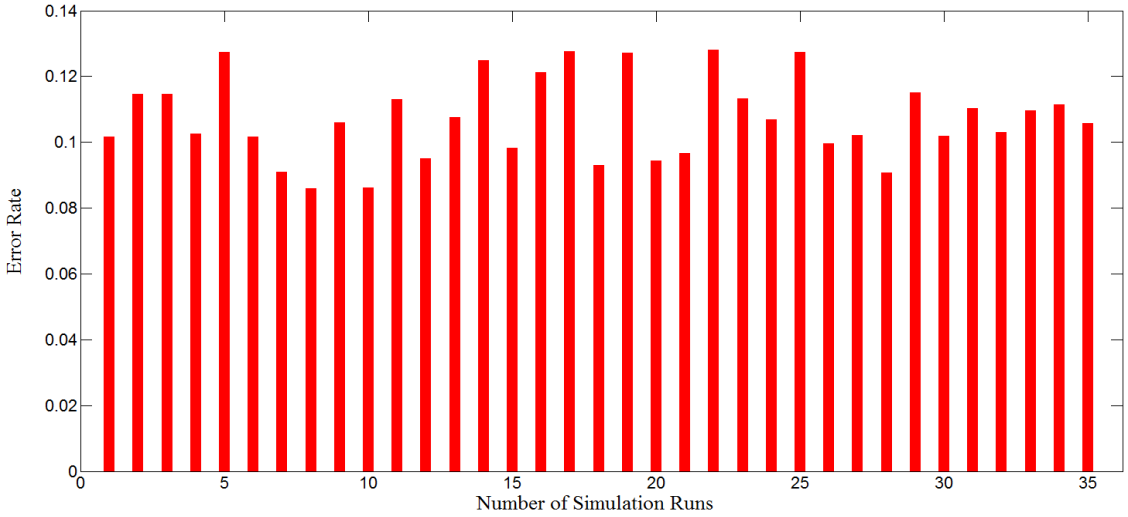

Figure 5: Misclassification of GP model based on 35 independent runs of the modified BUS framework

5 Conclusion

This paper has presented an implementation of the modified BUS framework as an inference method for GP classifiers. The modified BUS algorithm has been shown to produce samples from the relevant posterior distribution. From the comparison with Expectation Propagation, scaled information scores and misclassification error rates the framework appears to perform comparably well. Future work includes extending the framework to more complex scenarios, in particular multi-class classification.

REFERENCES

[1] M.R. Neal, *Probabilistic Inference Using Markov Chain Monte Carlo Methods*,Technical Report CRG-TR-93-1, Department of Computer Science, University of Toronto, 1993.

[2] C. Bishop, *Pattern recognition and Machine Learning*, Springer, 2006.

[3] T. Minka, *A Family of Algorithms for Approximate Bayesian Inference*,PhD thesis, Department of Electrical Engineering and Computer Science, MIT, 2001b.

[4] A.Smith, A.Gelfand, *Bayesian Statistics without tears: A sampling-resapling perspective*, American Statistical Association, **46(2)**, 84-88, 1992.

[5] D. Straub, I. Papaioannou, *Bayesian Updating with Structural Reliability Methods*, Journal of Engineering Mechanics, **141(3)** , 2015.

[6] S.K. Au, J.Beck, *Estimation of Small Probabilities in High Dimensions by Subset Simulation*, Probabilistic engineering Mechanics, **16**, 263-277, 2001.

[7] F.A DiazDelaO, A. Garbuno-Indigo, S.K Au, I. Yoshida, , *Bayesian Updating and Model Class Selection with Subset Simulation*,Computer Methods in Applied Mechanics and Engineering, **317**, 1102-1221, 2017.

[8] *Ionosphere Data Set*, UC Irvine Machine Learning Repository 1989.

[9] K. Zuev *Understanding the Subset Simulation Method for Rare Event Simulation*, Encyclopaedia of Earthquake Engineering, 2015.

[10] C.E. Rasmussen, C.K.I Williams, *Gaussian Processes for Machine Learning*, MIT Press, 2006.

[11] M. Kuss, C.E. Rasmussen, *Assessing Approximate Inference for Binary Gaussian Process Classification*, Journal of Machine Learning Research, **6**, 1679-1704, 2005.

[12] H.Nickisch, C.E. Rasmussen, *Approximations for Binary Gaussian Process Classification*, Journal of Machine Learning Research, **9**, 2035-2078, 2008.