Bayesian estimation of a probit regression model

Computational Statistics Project - Group: The Random Variables

Group members: Niccolo Diana, Rocco Gazzaneo, Silvia Juzova, Fabian Kraus

Please note that the derivation of the full conditionals are provided in the separate PDF file

The aim of this report is to briefly describe and discuss findings concerning the implementation of two algorithms (Metropolis Hastings and Auxiliary Gibbs) in the context of the bayesian estimation of a Generalized Linear model.

The idea behind the approaches is to generate a Markov Chain that, under the assumption of irreducibility and aperiodicity, enjoys ergodic properties by which its stationary distribution is the target distribution of the parameters of the linear model one wants to estimate.

In particular, the **link function** between the linear predictor and the response variable has been determined to be the *probit link*, which means that the residuals are assumed to follow the cumulative distribution function of a Standard Normal distribution.

Both the algorithms are implemented on simulated normal covariates, with pre-determined true beta values, upon which the random bernoulli are simulated by means of the probit link.

Metropolis Hastings

One prerequisite to run the MH algorithm is to determine a prior distribution. The algorithm was run using two different priors which output very similar results.

1. The first one we assumed it to follow a normal distribution, with zero mean and unit variance.
2. The second one follows a Uniform distribution between -500 and 500, a flat non informative prior.

This prior, together with the likelihood, will provide the functional form of the posterior distribution which will only allow **evaluation** but not **sampling** given the analytical intractability.

The betas are sampled from the proposal distribution recommended to be a multivariate normal distribution centered at the current beta, having as covariance matrix the fisher information evaluated at the current beta, where

$$I(\beta) = X'WX$$

W is diagonal with

$$W_{ii} = \frac{1}{V(Y_i)}\left(\frac{\delta\eta_i}{\delta\mu_i}\right)^{-2} = \frac{1}{F(x_i\beta)}\left(\frac{\delta F(x_i\beta)}{\delta x_i\beta}\right)^{2} = \frac{f(x_i\beta)^2}{F(x_i\beta)}$$

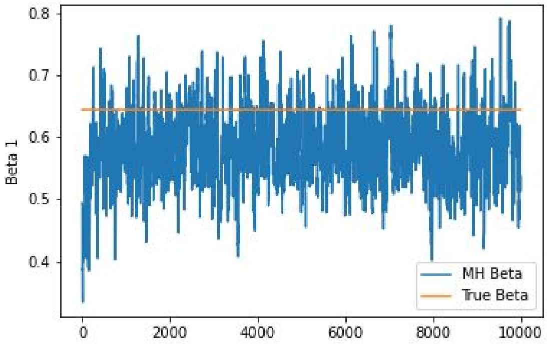
At each iteration the acceptance ratios between the posterior evaluated at the proposed beta and the posterior evaluated at the last accepted beta are around **33%** and **33.6%** with the normal and uniform priors respectively.

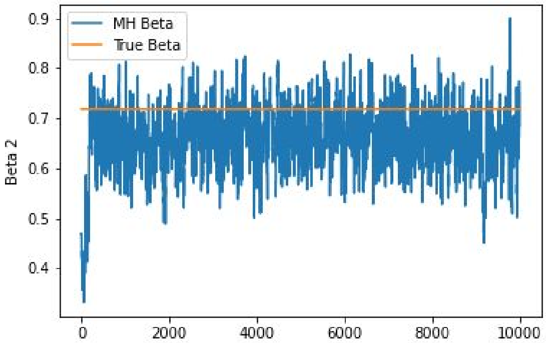
To get a rough sense of whether our algorithm is functioning properly, we could compare the true betas of our model against the betas produced by the last iteration of the algorithm. And we do so here below:

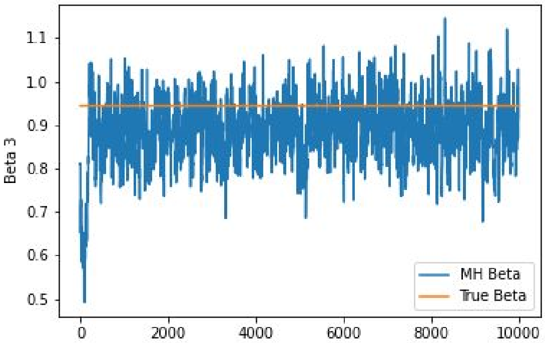
	β_1	β_2	β_3	β_4	β_5
True β	0.64405961	0.71851867	0.94386699	0.17103461	0.18710195
M.H. β	0.51364773	0.68417745	0.8701519	0.19248048	0.16533276

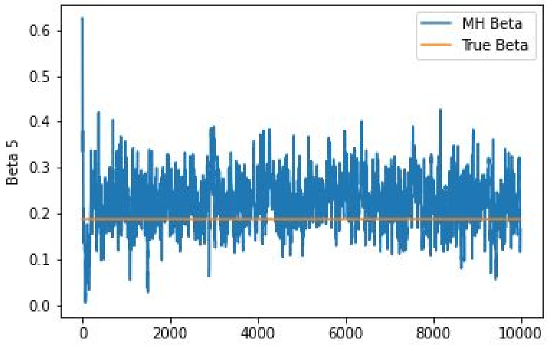
And at first glance they do seem like values generated by the same distribution. But let us investigate the process furtherly, with more exhaustive diagnostic checks.

Some trace plots will allow us to check for the convergence in distribution to the true model, and of course we must account for the initial "burn-in" period, thus we shall not take into account the first couple hundred iterations or so. Below we show the behavior of different betas drawn over 10,000 iterations of the algorithm.



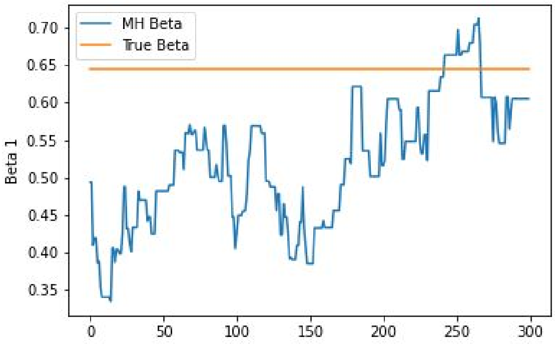


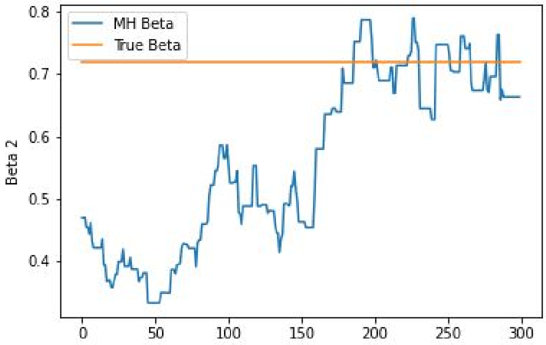


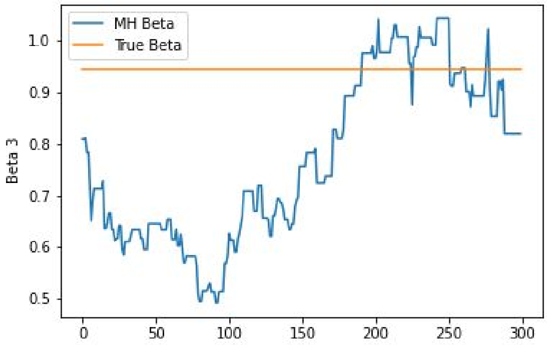


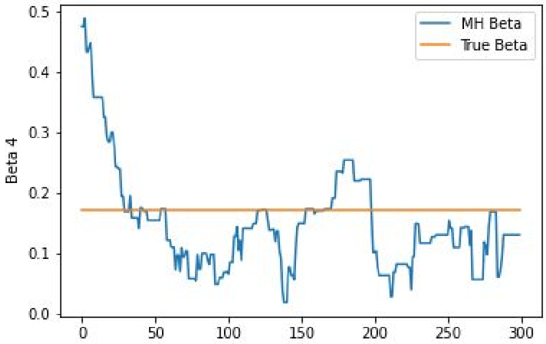
Indeed we can observe that the MH betas wiggle fairly close to the true betas when the number of iterations increases. Although it should be noted that even if the number of iterations can be considered sufficiently large for other algorithms, there are two other factors that must be accounted for in this specific case, and they both partly explain the contamination of our final results. First, unlike the Gibbs sampler, our acceptance rate is not 1, therefore most proposals of the betas are actually rejected (in our case since the acceptance rate is around 33%, approximately 67% of the betas proposed are being rejected by the algorithm), increasing the total number of iteration needed for the algorithm to achieve a significant result.
Secondly, it is known that the MH algorithm performs better as the number of parameters to be estimated decreases, thus it is an algorithm preferred in low-dimensional settings, suffering from an increase in the number of parameters.

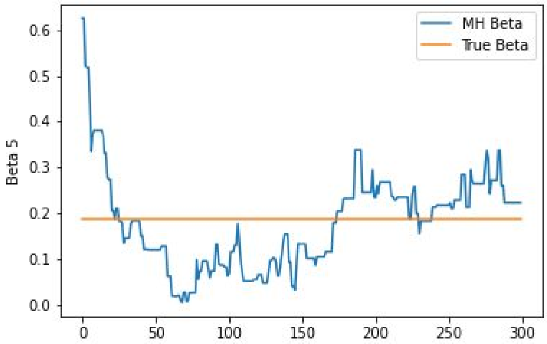
Thus plotting the burning period (in this case observed to be around 200 iterations) will allow us to check graphically the effect of the acceptance rate being around 33%, and also the fact that the betas sampled are exploring the parameter space, and after some exploration they end up wiggling around the value of the true beta.











As we can clearly see from the graphs, the “horizontal lines” drawn by the MH Beta plot indicate that the next beta proposed by the algorithm has been rejected, and therefore $\beta_{i;t+1}$ was taken to be the same as $\beta_{i;t}$.

Finally, the graphs show that the frequency of “horizontal lines” across the first 300 iterations is not high which is a sign that the algorithm did not get stuck into a value of beta and that it is freely able to explore the support, which we know it is an important assumption for the markov chain to converge to a stationary distribution (irreducibility).

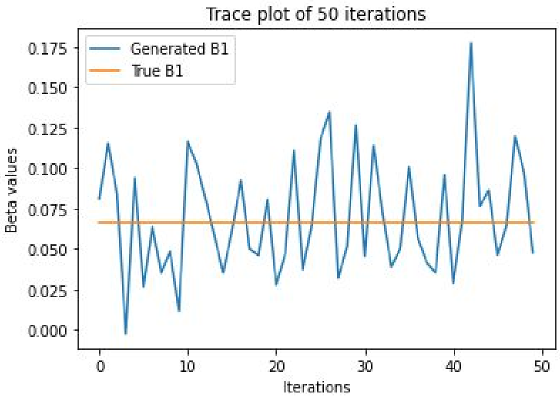
Auxiliary Gibbs Sampler

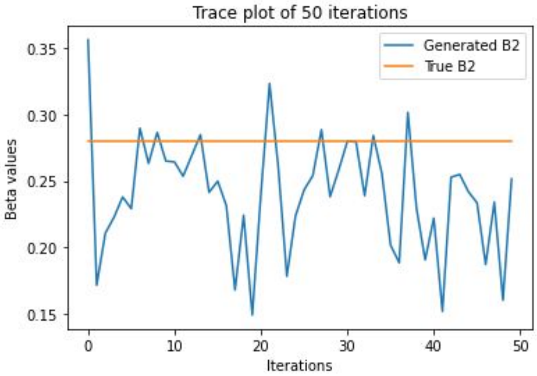
The following table summarizes the values assigned for the true betas that are to be approximated using the Auxiliary Gibbs Algorithm. When it comes to the sampling strategy, random selection from univariate distribution was implemented.

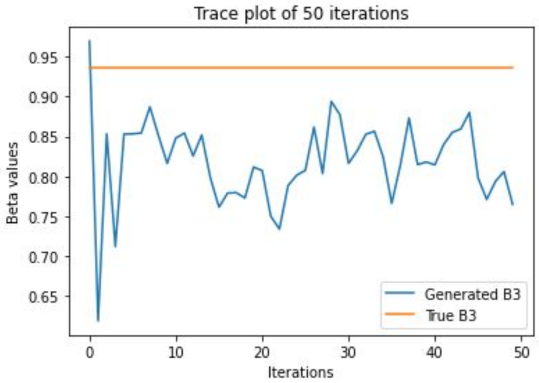
β_1	β_2	β_3	β_4	β_5
0. 06630875	0.27959659	0. 9355304	0. 47859363	0.14018002

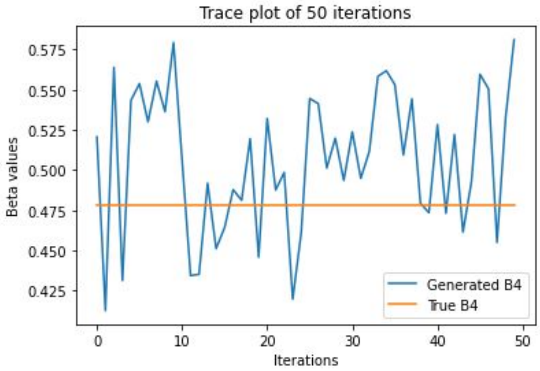
Betas with proper conjugate prior assignment

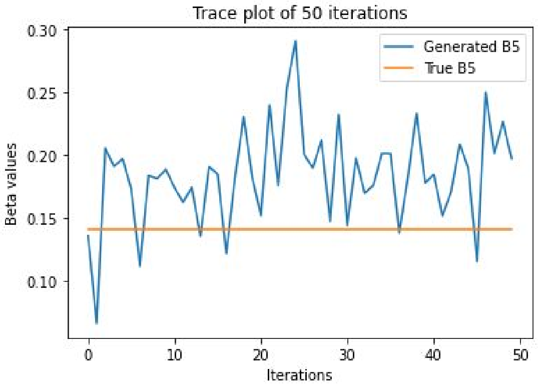
Before outlining the estimates of betas obtained from the approximation algorithm, initial tests on the correctness of the approach implemented are performed. The below graphs depict the trace plots obtained for the first 50 iterations of respective betas approximations. The indication of correct implementation of an algorithm is derived from the behavior of the sample path. A chain is considered to be mixing well, if the path moves quickly from the initial value and subsequently starts to wiggle around the region supported by the true value of beta. Thus, such a test can assess the capability of the sampling algorithm to explore a high share of the support of the true value.





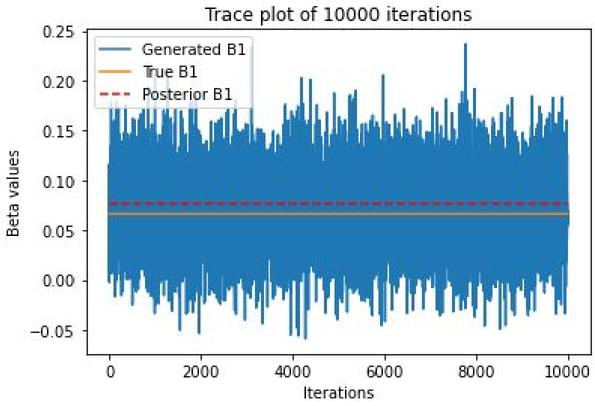


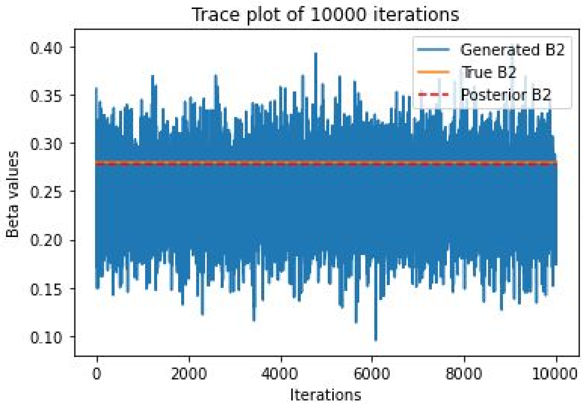


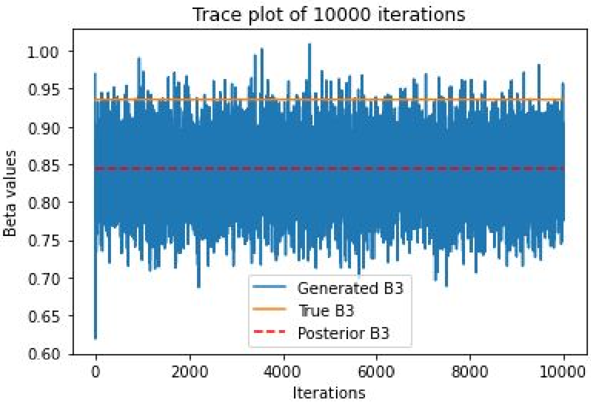


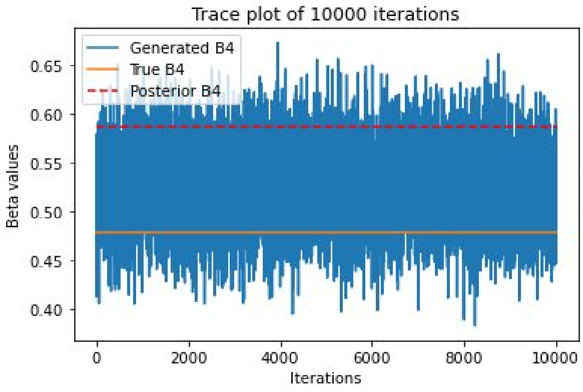
While indeed it can be observed that all chains of betas values do move promptly from their respective starting value, some betas are more successful than others at reaching the region supported by the corresponding true beta around which they continue to wiggle around. Given the underlying nature of Gibs algorithm, such a behavior is not surprising since the convergence towards the true value, especially in a high dimensional setting, generally requires a substantially long period. However, the results obtained give a spotty indication towards the correctness of the approach in place.

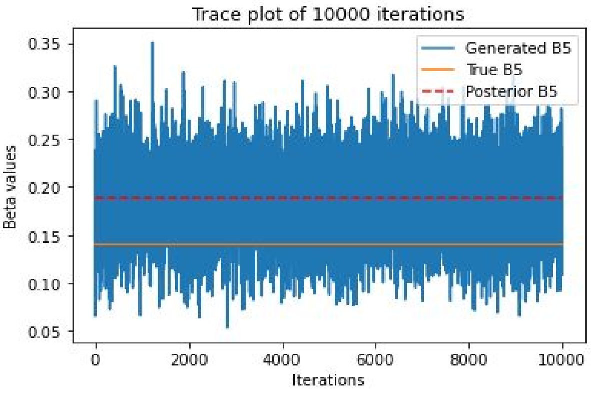
After performing the initial tests, the analysis further focuses on evaluation of a behavior of the respective chains over 10,000 iterations. It is believed that such a considerable period of chain computation should bring about the estimation of betas approximately close to their true value, while the individual chains should display a behavior of consistent wiggling around the support region of the corresponding true betas. Following trace plot graphs summarize the behavior of beta approximations across iterations.









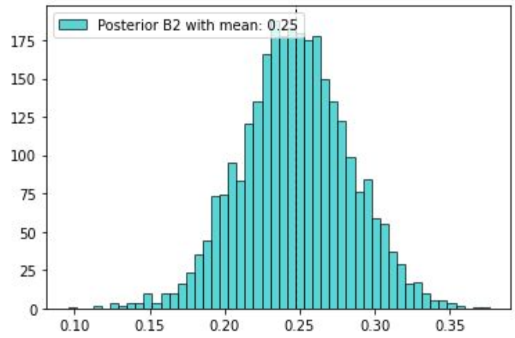


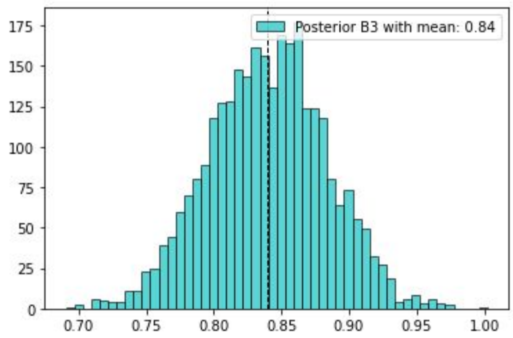
Additionally, the table below summarized the posterior approximation of individual beta coefficients.

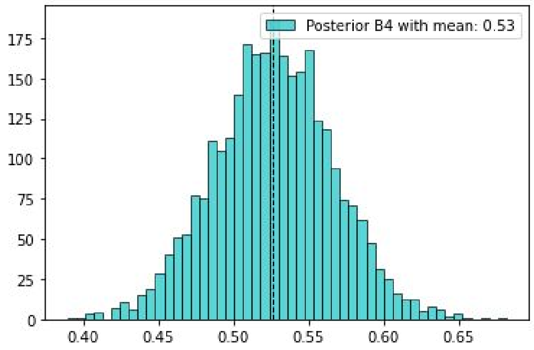
β_1	β_2	β_3	β_4	β_5
0. 07641629	0. 27808332	0. 84399948	0. 58669624	0. 18832157

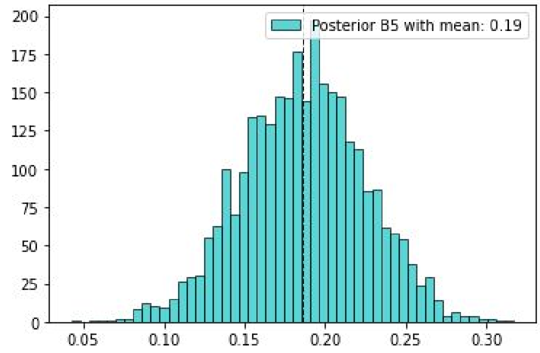
Observing the results obtained, the graphs provide a mixed conclusion on the correctness of the algorithm implemented. While the posterior approximation of values β_1 β_2 and β_5 are considerably close to the true values (with approximation error of 0.0101, 0.0015 and -0.0482 respectively), the values obtained for the rest of the coefficients differ more drastically from the true counterparts. Nevertheless, the posterior values of β_3 and β_4 could be still considered to be within acceptable regions given the underlying nature of and precision issues with Gibbs algorithm. The further discussion on performance of the algorithm and its comparison to MH algorithm can be found in the final section.

Additionally, an assessment of whether an i.i.d convergences can be reached is undertaken, where after 1,000 iteration of the algorithm every 10th observation of the chain is stored. Since the process is Markovian chain, we cannot simulate i.i.d sample while considering all the observations since the assumption of independence cannot be satisfied. It is possible to plot a distribution of such samples drawn and infer whether the betas collected are centering around the support region of corresponding true beta value.



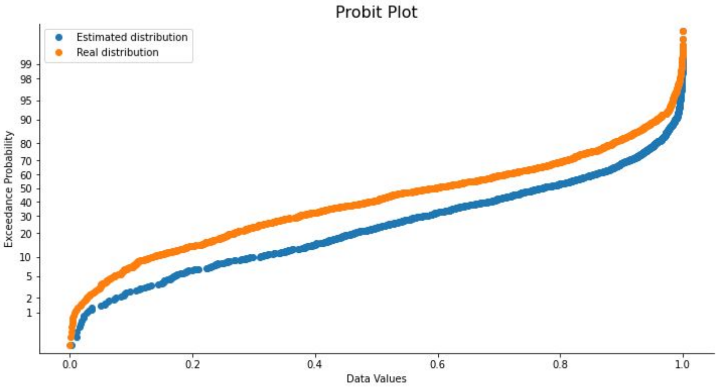






By looking at the graphs it can be concluded that the samples drawn for each beta display a behavior of a normal distribution, which is an expected result given the law of large numbers. The respective means associated with each coefficient are very close to the values obtained for betas after 10,000 iteration. An interesting observation might be the fact that the estimation error for β_4 decreases substantially if one would consider its average value rather than the value obtained at iteration 10,000. Such a behavior could be explained by the fact that as the chain wiggles around the region of acceptance, the process might have found itself a further away from the true value at the point of algorithm termination by pure coincidence.

Lastly, in aspiration to assess the predictive strength of the posterior betas obtained from the Gibbs sampler, such betas can be utilized in calculating a newly estimated probability values of the target binomial distribution using sum product between the estimated betas and the sample of original X variables. The estimated probability values can be plotted against the true probability values using a probit plot to understand whether the posterior model correctly approximates the true values of dependent variables in the target distribution.

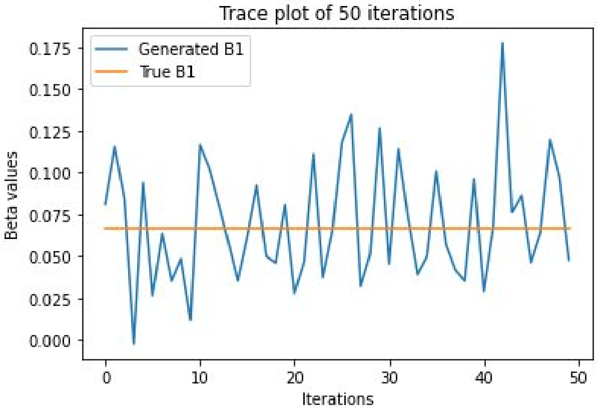


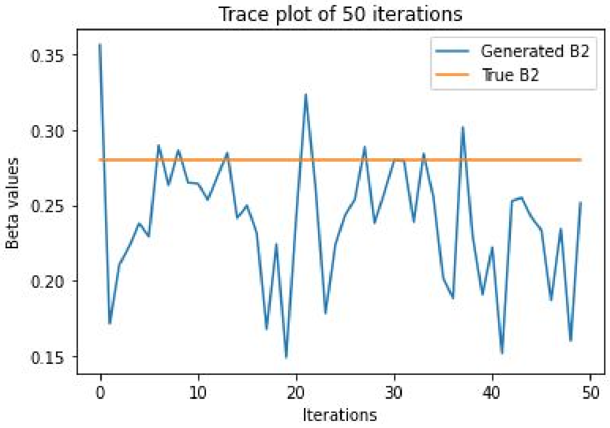
From the graph above it can be deduced that while the estimated distribution seems to have a very similar shape to the real distribution, the approximate model slightly underestimates the probability of acceptance across all values, while a higher error spread can be found for low values of p probability.

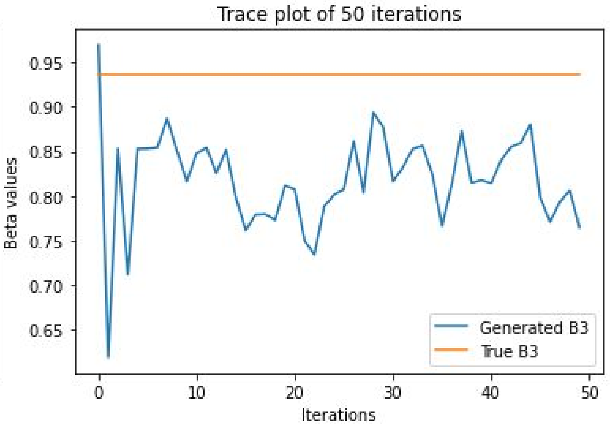
Betas without proper conjugate prior assignment

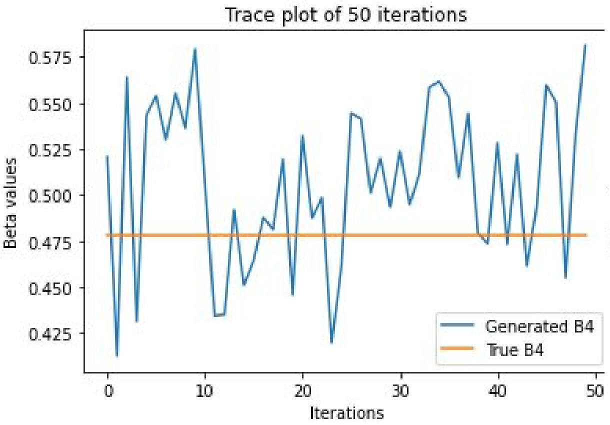
Furthermore, the Auxiliary Gibbs sampler is reused on the same true betas as outlined in the previous section, however the difference is now the assumption of proper conjugate prior, which is no longer assumed to be true.

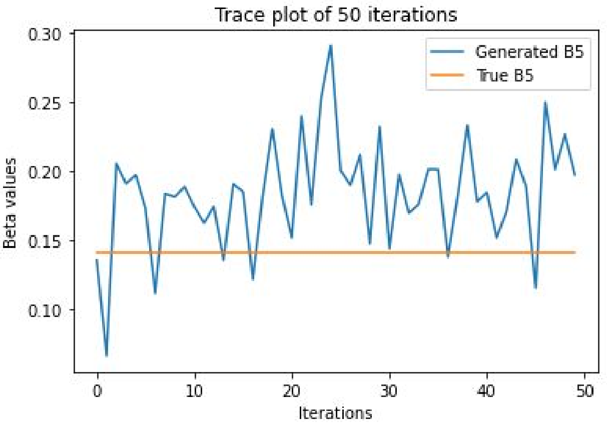
Analogously to the reasoning outlined in the previous section, graphs summarizing the behavior of beta chains are provided over the first 50 iterations.





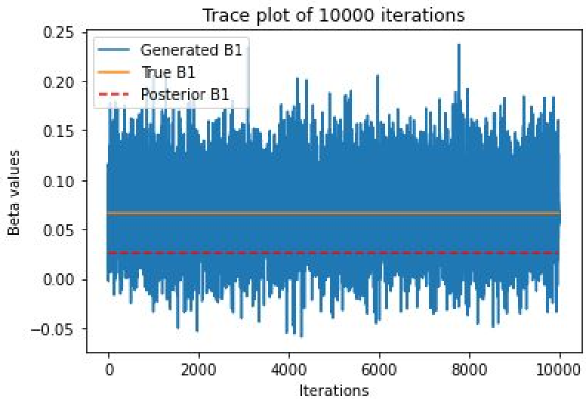


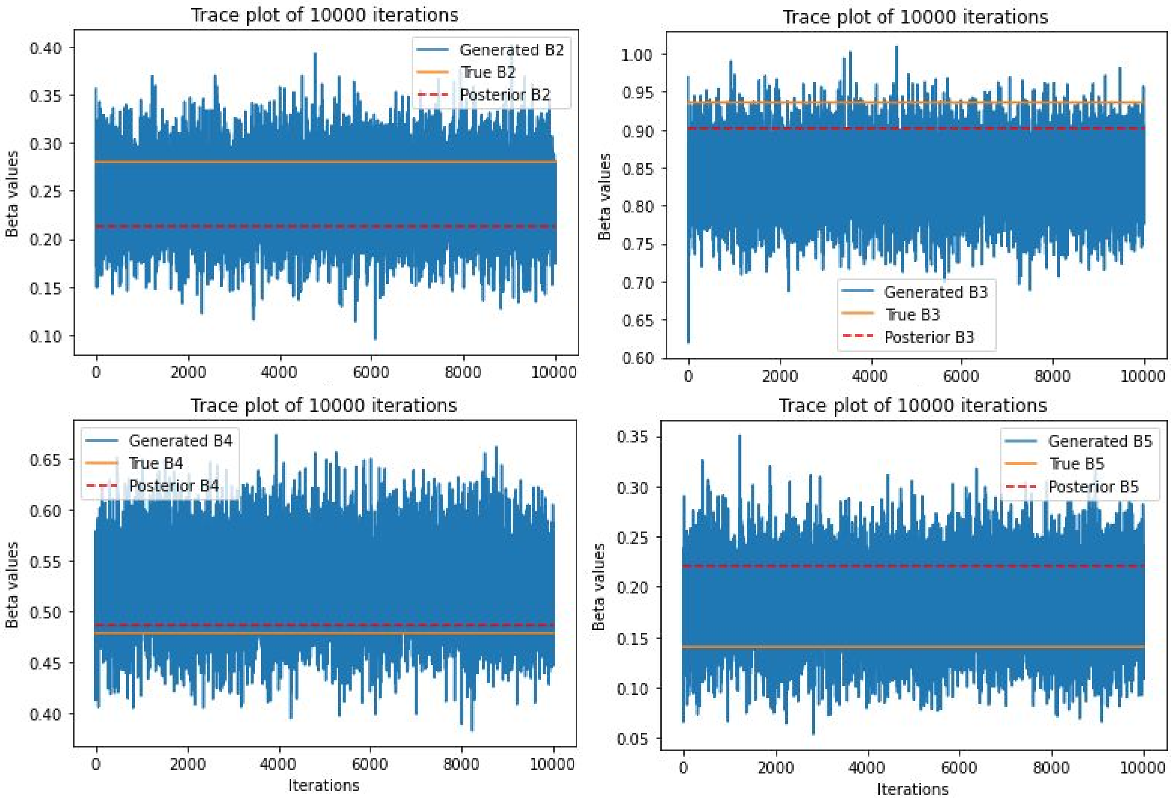




When comparing the result to the results obtained in the approximation using proper conjugate prior, it can be concluded that the behavior of each chain is almost identical to their counterparts in the previous section and thus the same conclusion can be reached as before.

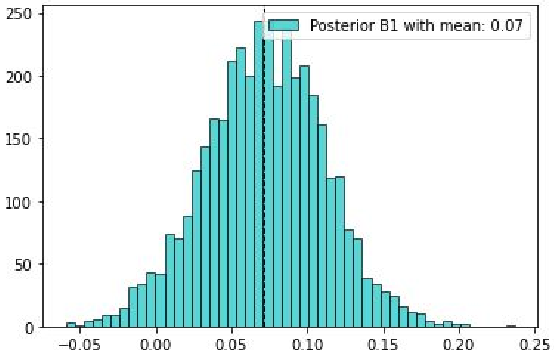
Furthermore, the sample chains are plotted across iterations of order 10,000 and the results are summarized in the following graphs as well as a table is provided with the summarized posterior values of respective betas.

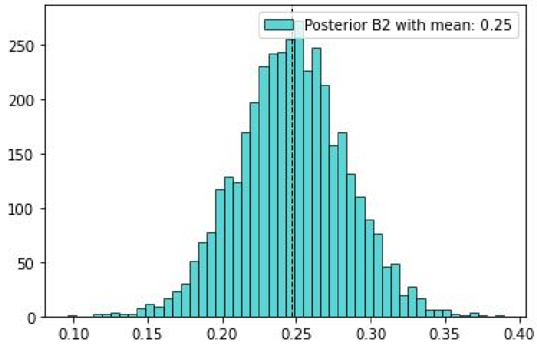


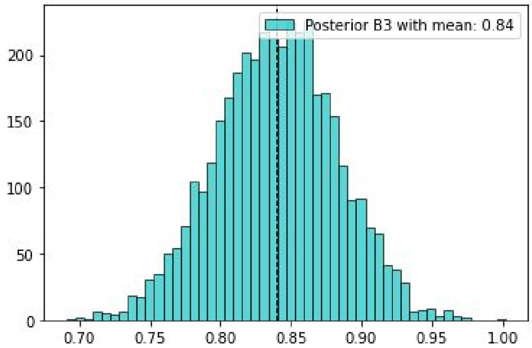


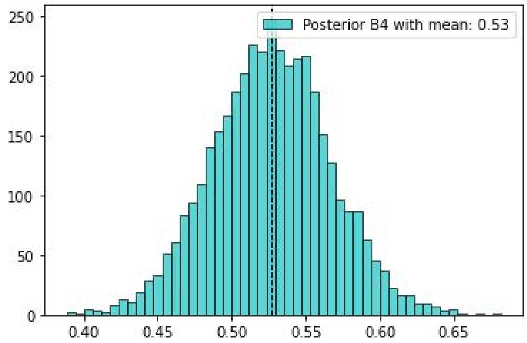
β_1	β_2	β_3	β_4	β_5
0. 02575516	0. 21307011	0. 90187839	0. 4861751	0. 22088558

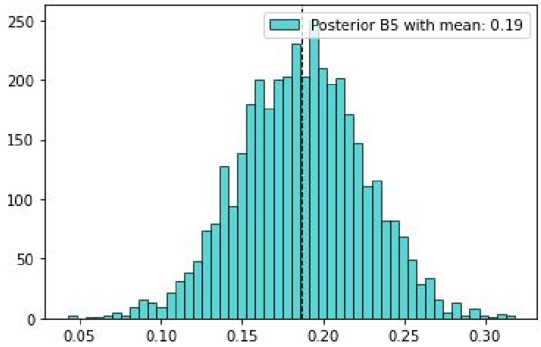
When analyzing the results obtained, one can notice a very intriguing remark. While the previous algorithm provided with considerably precise estimates of β_1 β_2 and β_5, the non-informative prior seems to work very well for estimates of β_3, β_4 (carrying approximation error of 0.0337 and -0.0076 respectively). Such results could be explained by the fact that as the chain wiggles around the region of acceptance, the process might have found itself closer to the true value at the point of algorithm termination by pure coincidence. This hypothesis is further confirmed by looking at the distribution of betas that are drawn as i.i.d samples from the iterations. On the other hand, it seems that for the remaining three coefficients the inclusion of informative prior seemed to help with precision, however such a claim can be again either confirmed or disproved by looking at histogram of their respective distribution. The following histograms summarize the findings.





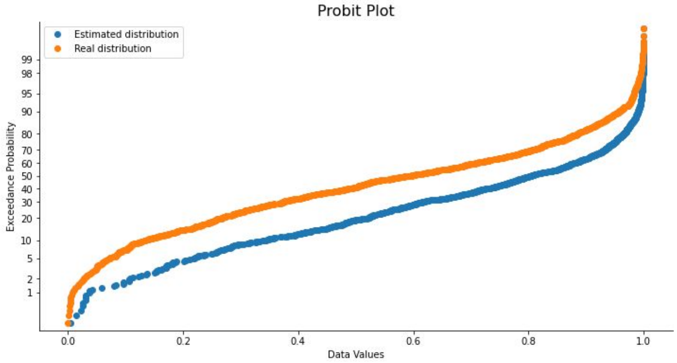






Observing the results, one can notice that the distributions and their average values are almost identical to the ones with proper conjugate prior. Therefore, it seems to be the case that inclusion of such prior did not increase the performance of the algorithm and the more precise estimates of values β_1 β_2 and β_5 could have been achieved by termination of process at point in time in which the algorithm's chain is at closer position to the true value in its wiggling. Additionally, it can be yet again concluded that the distributions of the respective betas display normality features due to the law of large numbers.

Lastly, following the same reasoning as in the section before, a probit function is plotted that examines the predictive ability of the posterior betas obtained.



Comparably to the conclusion reached in the previous section, the distribution of the estimates displays the same shape as their true counterparts, however the probability of acceptance is underestimated across all values of p. Nevertheless, after careful observation of the graph, one can notice that the error spread between the estimated and the true probability values is more serious than in the approach using informative prior. This gives an indication for preference of approach that considers prior beliefs.

Performance Comparison: Metropolis Hastings vs Gibbs

To compare the performance of the two algorithms we should take into account **efficiency** and **efficacy** of the two. We could start by comparing the two running times of the algorithms fixing the same random seed (meaning data fed into the algorithm in our case), number of parameters to be estimated and iteration of the algorithm. To do so, we've fixed the number of parameters to be equal to 5 plus an intercept, we have used the same random seed and we have set the number of iterations to be 10,000. The approximated results expressed in seconds were the following:

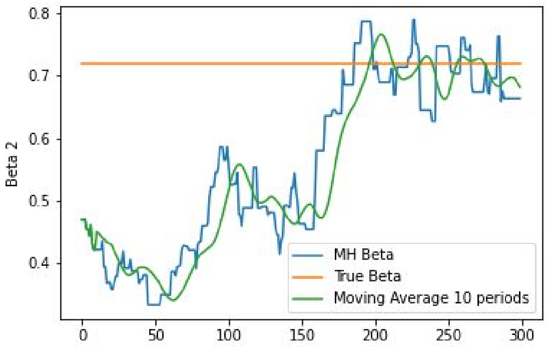
Time elapsed 10k iterations MH: 98.51790 seconds

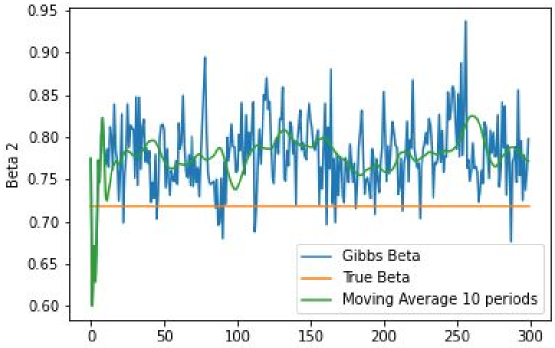
Time elapsed 10k iterations Informative Gibbs: 8.99834 seconds

Time elapsed 10k iterations Noninformative MH: 11.87153 seconds

Note that Gibbs holds very similar results when run with informative and noninformative prior, whereas if we consider Gibbs against MH, we can see a significant difference in running time between the two. In fact, Gibbs is approximately 98/10 times faster than Metropolis Hasting given the conditions reported above. This finds an explanation again in the way the two algorithms are constructed, and because we have decided to opt for a fast immediate updating of parameters in the case of Gibbs.

For what concerns the number of iterations needed for the algorithm to start sampling from the "true distribution", we have significantly different results between the two. We illustrate them graphically below (we make use of β_2 as an example, but very similar results hold even for the others):





On the left we have the Metropolis Hastings' first 300 iterations, and on the right we have Gibbs' first 300 iterations. We make use of moving averages of 10 periods to highlight how MH resulted to be slower in its convergence (longer "burn in" period), yet more precise when we look at the moving average. On the other hand, Gibbs has shown superior initial speed in its parameter space exploration, yet it suffers when it comes to its precision.