university of groningen

faculty of science and engineering

The Impact of Class-Based Noise on Bayesian and Frequentist Logistic Regression

Mohammad Siam Shahkhan

university of groningen / faculty of science and engineering

University of Groningen

The Impact of Class-Based Noise on Bayesian and Frequentist Logistic Regression

Bachelor Thesis

To fulfill the requirements for the degree of
Bachelor of Science in Mathematics
at University of Groningen under the supervision of
Prof Dr. M.A. (Marco) Grzegorczyk (Mathematics, University of Groningen)
and
Dr. W.P. (Wim) Krijnen (Mathematics, University of Groningen)

Mohammad Siam Shahkhan (s4060792)

August 9, 2023

Contents

Abstract

Real world data often contains noise, thus, in classification tasks it is important that the algorithms used are robust against noise. In this research, the aim is to compare the performance of Frequentist and Bayesian methods, specifically when it comes to this fundamental issue; handling noisy data sets. In regard to the Frequentist method, we will be looking at AIC or Akaike information criterion and BIC or Bayesian information criterion. For the Bayesian approach, Metropolis-Hastings MCMC and Reversible Jump MCMC (RJMCMC) Processes are evaluated. The data set used is a benchmark data set; Breast Cancer Wisconsin Diagnostic data set that has binary labels for classification. The algorithms are trained and tested on the data set. Furthermore the accuracy of these algorithms are compared against the increasing class-based noise levels in the training data. In addition to that, the classification threshold will also be changed to observe its effects. This study shows that under increasing class-based noise RJMCMC performs with the best accuracy. A significant drawback of the RJMCMC algorithm is its computational complexity when contrasted with the GLM stepwise AIC and BIC procedures. While the current study focused on specific noise levels and data sets, future work could explore different noise structures.

1 Introduction

A very peculiar phenomenon occurs when two people look at the same object. With nothing different in the superficial visage of the object, the two people can perceive a different view, fully formed and burned into their retinas. For instance, one may see a glass half full whilst the other a glass half empty. An extension of this idea can be seen when statisticians look at data.

One statistician may try to make sense of data by using their intuition, assuming that the parameters of the data are fixed. They view probability as the long run frequency of an event occuring in repeated experiments. In this case, probability is seen as the limit of the relative frequency in many trials and there is confidence that as long as many identical experiments are repeated, the ”true” value will emerge asymptotically. This approach refers to the Frequentist way of looking at data. On the other hand, another statistician may believe that probability is subjective and based on prior beliefs or uncertainty. In this statisticians viewpoint, this is the Bayesian way. In Bayesian inference, guided by the well-known Bayes theorem, prior beliefs are incorporated and updated to the direction of evidence.

These two approaches are the two major frameworks for statistical analysis. While both approaches have their own strengths and weaknesses, one of the major differences between them lies in how they deal with uncertainty. One way to describe uncertainty in statistical analysis is the concept of noise. In statistical analysis, it is fundamentally important to classify data in a consistent way.

In complex data sets of the modern world, noise poses a challenge to this effort of consistency, and thus, the reliability of statistical analyses. When it comes to accounting for noise, several decisions are to be taken in regard the direction and methods of the analysis. A great deal of scientific effort is undertaken to ensure clean useful data sets are available for training and testing purposes.

In this research, the aim is to compare the performance of Frequentist and Bayesian methods, specifically when it comes to this fundamental issue; handling noisy data sets. In particular, both frameworks will be compared on the model of logistic regression. Logistic regression is a widely used model in classification tasks and finds its way in many modern applications which is why this study chooses to discuss the effects of noise on it. The aim is to train and test methods from both frameworks on a data set and to compare the accuracy with regards to the noise level.

When deciding on the breadth of approaches there are in both the frequentist and Bayesian toolbox, this study has narrowed down four different inferences. In regard to the Frequentist method, we will be looking at AIC or Akaike information criterion and BIC or Bayesian information criterion. For the Bayesian approach, two types of Markov Chain Monte Carlo (MCMC) Processes are evaluated. Both of these processes will be judged on a widely known benchmark data set; Breast Cancer Wisconsin (Diagnostic) Data Set.

1.1 Research Questions

To summarize, this thesis focuses on the following problems:

Q1. What is the impact of different levels of class-based noise on the accuracy of both approaches on logistic regression?

Q2. What are the advantages and disadvantages of both approaches and their respective algorithms?

2 Background Literature

In this section the technical details behind logistic regression will be explained. Logistic regression is the prediction model of choice, as our data set has binary outcomes, making it it ideal for logistic regression to be trained and tested on. Since I will be looking at models from both statistical approaches, I will attempt to clarify the underlying structure in these models and how model selection works within both of these frameworks. For both frameworks, two models have been selected, each quite similar to one another. In this section the technical details behind each model and its purpose are discussed.

2.1 Logistic Regression

Similar to linear regression, logistic regression models the relationship between the dependent variable and the independent variables. However, in this case, the model is used to make a prediction on a categorical variable instead of a continuous one. In Binary logistic regression we look at two outcomes where the response variable Y can be either 0 or 1. This kind of modeling is also known as classification. [1] A logistic model in simple terms models the probability of an event taking place by having the log-odds for the event be a linear combination of one or more independent variables. To formulate the logistic model in terms of regression, statisticians devised a linear relationship between the logistic model and the outcome variables.

$$\log \frac{\pi_i}{1-\pi_i} = \beta_0 + x\beta$$

Then in solving for π_i we would get

$$\pi_i = \frac{e^{\beta_0 + x\beta}}{1+e^{\beta_0 + x\beta}} = \frac{1}{1+e^{-(\beta_0 + x\beta)}}$$

In order to minimize the risk of miss-classification, we predict $Y = 1$ when $p \geq 0.5$ and $Y = 0$ when $p < 0.5$. Thus in this way the model gives us a linear classifier.

The linear logistic regression model $\log \frac{\pi_i}{1-\pi_i} = \beta_0 + x\beta$ is a special case of the general logistic regression model

$$logit\pi_i = log\frac{\pi_i}{1-\pi_i} = x_i^T\beta$$

where x_i is a vector of covariates while β is the parameter vector. This method is very powerful and effective for analysis of data involving binary or binomial responses and several covariates.

As in linear regression, the maximum likelihood estimates of the parameters β and the probabilities are obtained similarly. We maximize the log-likelihood function $l(\pi; y) = \Sigma_{i=1}^{N}[y_i \log \pi_i + (n_i - y_i)\log(1-\pi_i) + log\binom{n_i}{y_i}]$. The MLE process has been omitted for the sake of brevity, assuming the reader is familiar with the concept.
However for the case of this study, we will estimate the parameters using goodness of fit statistics. The Akaike information criterion (AIC) and the Baysesian information criterion (BIC) are two goodness of fit statistics based on using the log-likelihood function. They account for the number of parameters estimated. The AIC is defined as follows

$$AIC = -2l(\hat{\pi}; y) + 2p$$

BIC is defined similarly

$$BIC = -2l(\hat{\pi}; y) + p \times \ln(\text{number of observations})$$

Despite some existing differing definitions, these definitions are used in the research. This is due to the choice of software used in this reserach, R, which also uses these definitions. [1]

2.2 Bayesian Logistic Regression

A quick primer on Bayesian statistics: Unlike in Frequentist statistics, unknown parameters are not assumed to be constants. Instead, they are assumed to be random variables themselves. Parameters have prior distributions and the posterior distribution has to be inferred from them and the likelihood. In Bayesian logistic regression, the likelihood function matches that of the Frequentist one. It quantifies the compatibility between the observed data and different parameter values. [2]

To obtain the posterior distribution, which represents our updated beliefs about the parameter(s) after observing the data, the prior distribution and the likelihood function are combined. The posterior distribution is proportional to the product of the likelihood function and the prior distribution, and it represents the distribution of the parameter(s), given the observed data. Following is how we derive the posterior distribution: $p(\theta)$ is the prior distribution of the parameter. The likelihood is $p(y_1,...,y_n|\theta)$ and the posterior distribution is

$$p(\theta|y_1,...,y_n) \propto p(y_1,...,y_n|\theta) \cdot p(\theta)$$

Where θ is the unknown parameter and y_i $i \in N$ is the observed data. This derivation of the posterior distribution follows from the famous Bayes' equation

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)}$$

$p(y)$ is the same for every value in the numerator thus it can be safely ignored. [3] The prior distribution $p(\theta)$ has to be specified in advance, it is not allowed to depend on the observed data. Typically, priors are what the analyst believes about the conditions. The priors could be a completely uninformed opinion, although this is uncommon. The priors can also incorporate information about the design of the study or the model. An example of this is a clinical trial study, where historically, the priors have been a certain distribution. The parameters of the prior distribution are known as hyper parameters.

In logistic regression, to incorporate Bayesian inference, prior distributions must be specified for the parameters β where $\beta = (\beta_0,...,\beta_k)^T$ is the vector of $k+1$ regression parameters. For the purposes of this study we assume that β is normally distributed $\beta_i \sim N(0,\sigma^2)$ $i = (0,...,k)$. For the posterior distribution we then have

$$p(\beta|y) \propto p(y|\beta) \cdot p(\beta)$$

Where $\beta = (\beta_0,...,\beta_k)^T$ $y = (y_1,...,y_n)^T$ and $p(\beta) = \prod_{i=0}^{k} p(\beta_k)$

However, in practice, obtaining the exact form of the posterior distribution can be analytically intractable for complex models. Thus, we have to use methods such as Gibbs sampling or the Metropolis-Hastings algorithm to draw samples from the posterior distribution. In this study, the latter will be used.

2.2.1 Markov Chain Monte Carlo

A Markov Chain is a sequence of random variables $X_0, X_1, ..$ taking values in a set S, called the state space. For every state $i, j \in S$ there are probabilities $P_{i,j} \in [0,1]$ such that the state moves from state i to j. The Markov Chain follows the so-called Markov property which is that

$$P(X_{t+1} = i_{t+1} | X_t = i_t, ..., X_1 = i_1) = P(X_{t+1} = i_{t+1} | X_t = i_t)$$

This means that the state at time $t+1$ depends only on the state at time t. This is called the "Memoryless property" [4] This definition is crucial in order to adequately explain the process of Markov Chain Monte Carlo methods.

Monte Carlo simulation is the process of repeated random sampling from a distribution to approximate complex mathematical expressions or solve problems that may not have an analytical solution. A Markov Chain Monte Carlo (MCMC) process combines elements of Markov chains and Monte Carlo methods to generate samples from a target probability distribution. In MCMC, the goal is to sample from a probability distribution, often a high-dimensional and complex posterior distribution. The MCMC process constructs a Markov Chain whose equilibrium distribution (stationary distribution) is the target distribution from which we want to draw samples. The chain is designed in such a way that after running it for a sufficient number of steps, the generated samples closely approximate the desired distribution. [4]

A very well known MCMC algorithm is the Metropolis-Hastings algorithm, proposed by Nicholas Metropolis in 1953 in the seminal paper *Equation of State Calculations by Fast Computing Machines*. The Metropolis-Hastings algorithm generates a sequence of sample values in a way that progressively improves the approximation to the desired distribution. These samples are obtained step-by-step, forming a Markov Chain, where each new sample depends only on the previous one. During each iteration, the algorithm proposes a new candidate sample based on the current value. Subsequently, the candidate is accepted with a certain probability, and if accepted, it becomes the next sample; otherwise, it's rejected, and the current value is reused for the next iteration. The acceptance probability is determined by comparing the function values, $f(x)$, of the current and candidate samples relative to the desired distribution.

3 Methods

3.1 Data Set

The Breast Cancer Wisconsin (Diagnostic) Data Set is a widely used data set in machine learning and pattern recognition. It was created by Dr. William H. Wolberg at the University of Wisconsin Hospitals and is publicly available from the UCI Machine Learning Repository. This data set contains features computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. These features are used to classify the breast mass as either malignant (cancerous) or benign (non-cancerous). The original dataset includes 699 instances, with each instance having the following attributes:

- ID: A unique identification number for each sample
- Diagnosis (target variable): The class label, where ”M” stands for malignant (cancerous) and ”B” stands for benign (non-cancerous).
- Ten real-valued features computed from the cell nuclei present in the image including radius (mean of distances from the center to points on the perimeter).

Researchers and practitioners often use this data set for building and testing machine learning models for breast cancer classification. The goal is to train a model that can accurately predict whether a breast mass is malignant or benign based on the provided features.

When using this data set for real-world applications, it is essential to consider that it might not fully represent the complexity and diversity of breast cancer cases found in larger data sets. The size of the data set is relatively small compared to some modern data sets. This may pose a problem in some classification algorithms. Nonetheless, it is viewed as a benchmark data set which is quite robust.[5]

3.2 Generating Noise

Our data set will be partitioned into training data and testing data, in a 70/30 split. In order to analyze the effect of noise, we will conduct training and testing with different levels of noise. We will begin by adding 10% of noise to the training set and increasing each time by 10% until we reach 70% noise in the training set. This percentage of noise reflects the instances of data that are ”corrupted” vs the clean instances.

The type of noise that introduced in this study is class noise. In in effect, it is the shuffling of class labels in the data set. For example, if an instance has a class label of ”M” for malignant, we will swap it to ”B” for benign and in this way, some instances will have incorrect labels.

3.3 Algorithms

In each of the four algorithms mentioned in this section, the training and testing is conducted over 50 iterations, after which the results are averaged out and plotted against the level of noise. In addition, the process on differing values of the π is tested to see how it affects the algorithms and the noisy data, as this is the classification probability. I will test π for values of 0.5, 0.6 and 0.7

Algorithm 1 Noisy Data Class Shuffling

Require: Training data $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, Noise percentage p.

1: Determine the number of instances to be modified: $n_{\text{noisy}} = p \times n$.
2: Randomly select n_{noisy} instances from D. Let this subset be D_{subset}.
3: **for** each (x_i, y_i) in D_{subset} **do**
4: Shuffle or replace the label y_i with a label from the set of possible labels, ensuring it's different from the original.
5: **end for**
6: The modified dataset D' now contains both original and noisy labels.

3.3.1 GLM stepwise AIC

To make matters simple, I have elected to use the prebuilt GLM logistic regression function found within R. This enables making modifications in the code to account for the stepwise AIC model selection. Using the GLM function in R, it is specified that the family parameter is binomial. First a null model is specified, and then a full model. One by one, variables will be added to observe how they affect the AIC score. The model with the lowest AIC score is then chosen and the noise levels are added.

Algorithm 2 Logistic Regression with stepwise AIC in R

1: **Input:** Noisy training data, test data
2: **Output:** Prediction accuracy
3: **Begin**
4: Randomly select a subset of data points to be corrupted.
5: Add noise to the selected data points by randomly shuffling their 'diagnosis' labels.
6: Fit a null and a full logistic regression model to the noisy training data.
7: Perform stepwise regression starting with the null model.
8: Predict the 'diagnosis' labels on the test data using the new model.
9: Convert predicted probabilities to class labels using a threshold of 0.5.
10: Calculate prediction accuracy by comparing predicted labels with actual labels.
11: **End**

3.3.2 GLM stepwise BIC

The smae procedure is repeated as before - however, there is an adjustment made to the AIC calculation by increasing the penalty for adding parameters as the sample size increases, with the penalty factor set to log(n) which is equivalent to using BIC.

```
stepwiseAIC <- stepAIC(..., direction = "both")
```

We have instead added

```
stepwiseBIC <- stepAIC(..., direction = "both", k = log(nrow(train_data)))
```

3.3.3 Metropolis-Hastings MCMC Algorithm

In this section, the Metropolis-Hastings algorithm is implemented for every level of noise. As established in the background section, the Markov Chain that is constructed has its transition density

represented as:

$$p(y|x) = q(x,y) \cdot A(x,y)$$

This is consistent with the Principle of Detailed Balance, which is described as:

$$\frac{p(y|x)}{p(x|y)} = \frac{q(x,y)}{q(y,x)} \cdot \frac{A(x,y)}{A(y,x)} = \frac{f(y)}{f(x)} = \frac{p(y)}{p(x)}$$

Given this, the constructed Markov chain reaches a stationary distribution with density $p(x)$ (for any x in the state space S). In simpler terms, letting this Markov chain run for an extensive duration ensures that it provides samples that come from the distribution P.

It's a method to produce samples from the posterior distributions. The form of these posterior densities is always known, and can be expressed as:

$$p(\theta|y_1,\ldots,y_n) \propto p(y_1,\ldots,y_n|\theta) \cdot p(\theta)$$

With the use of the Metropolis-Hastings MCMC approach, a Markov chain can be formulated that, upon convergence, aligns with the stationary distribution. In the context of this study, this distribution is the posterior of:

$$\theta|(Y_1 = y_1,\ldots,Y_n = y_n)$$

[6]

Algorithm 3 Metropolis Hastings MCMC Logistic Regression

Initialize an initial state $\beta_0^{(0)} = 0, \ldots, \beta_k^{(0)} = 0$.
for $t = 1,2,\ldots,T$ **do**
  for $i = 0,\ldots,k$ **do**
    Sample $\varepsilon_i \sim UNI([-v,v])$
    Propose to replace the current regression parameter vector:

$$(\beta_0^{(t)},\ldots,\beta_{i-1}^{(t)},\beta_i^{(t-1)},\beta_{i+1}^{(t-1)},\ldots,\beta_k^{(t-1)})^T$$

    with:

$$(\beta_0^{(t)},\ldots,\beta_{i-1}^{(t)},\beta_i^{(t-1)}+\varepsilon_i,\beta_{i+1}^{(t-1)},\ldots,\beta_k^{(t-1)})^T$$

    Compute:

$$A = \min\left(1, \frac{p_n(\beta_i^*)\cdot p(\beta_i^*)}{p_n(\beta_i^{(t-1)})\cdot p(\beta_i^{(t-1)})}\right)$$

    Sample $u \sim \text{Uniform}(0,1)$
    if $u \leq A$ **then**
      Accept β^* and set $\beta^{(t)} = \beta^*$
    else
      Set $\beta^{(t)} = \beta^{(t-1)}$
    end if
  end for
end for
return the sequence $\{\beta^{(t)}\}_{t=0}^T$

3.3.4 Reversible Jump MCMC Logistic Regression

Similar to the Metropolis-Hastings MCMC algorithm, RJMCMC adds an additional step at the start to ”jump” between models.

Given a set of competing models $M = \{M_1, M_2, \dots\}$, each model M_k has its own posterior distribution defined as:

$$p(\theta_k|y,k)$$

where $p(y|\theta_k,k)$ is the probability model and $p(\theta_k|k)$ is the prior for the parameters of model M_k.
The goal of the RJMCMC method is to perform simulations that move between different models, each with varying dimensions. This jumping between models results in samples from a joint distribution $p(\theta_k,k)$. The methodology ensures reversibility to maintain balance in the Markov Chain, guaranteeing convergence to the desired distribution.
[7]
The RJMCMC algorithm, given a current state (k,θ_k), can be described in the following steps:

1. **Proposal Step**: Propose moving to a new model $M_{k'}$ based on a certain probability $J(k \to k')$.
2. **Sampling Step**: Sample a value u from a proposal density that depends on the current model and the proposed new model.
3. **Dimension Matching**: Transform the sampled value and the current state using a bijective function $g_{k,k'}$. This step ensures that dimensions match between the old and new models.
4. **Acceptance Step**: Calculate the acceptance probability for the new model using a formula that involves the ratio of the posterior probabilities of the old and new models and the ratio of the proposal densities.

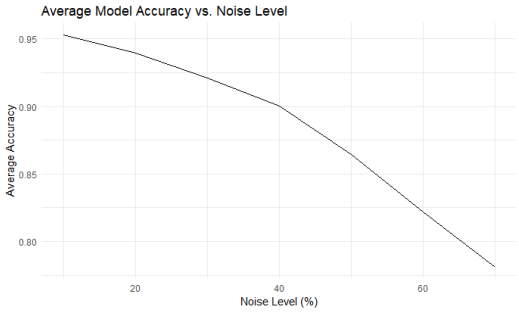
By repeating these steps, a sample set for the model indicators is obtained, from which we can estimate the probability $Pr(k|y)$. This methodology allows for jumps between models of different complexities, facilitating model selection and parameter estimation simultaneously. [7]

Algorithm 4 Reversible Jump MCMC Logistic Regression

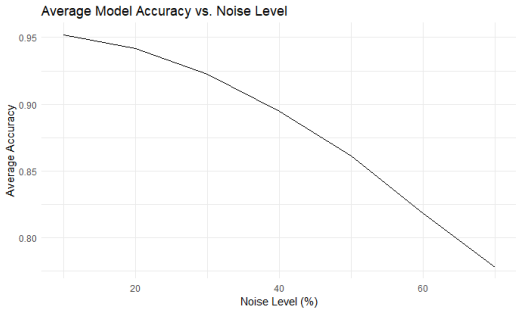
RJMCMC Move Iteration:
Let current indicator vector be $v^{(t)} = (v_1^{(t)}, \ldots, v_k^{(t)})$
Let current regression parameter vector be $\beta^{(t)} = (\beta_0^{(t)}, \beta_1^{(t)}, \ldots, \beta_k^{(t)})^T$
Randomly select a covariate $j \in \{1, \ldots, k\}$.
if $v_j^{(t)} = 1$ **then**
    Propose move to:
    $v^{(*)} = (v_1^{(t)}, \ldots, v_{j-1}^{(t)}, 0, v_{j+1}^{(t)}, \ldots, v_k^{(t)})$
    $\beta^{(*)} = (\beta_1^{(t)}, \ldots, \beta_{j-1}^{(t)}, 0, \beta_{j+1}^{(t)}, \ldots, \beta_k^{(t)})^T$
else if $v_j^{(t)} = 0$ **then**
    Sample $\beta_j^{(*)} \sim \mathcal{N}(0, \sigma^2)$
    Propose move to:
    $v^{(*)} = (v_1^{(t)}, \ldots, v_{j-1}^{(t)}, 1, v_{j+1}^{(t)}, \ldots, v_k^{(t)})$
    $\beta^{(*)} = (\beta_1^{(t)}, \ldots, \beta_{j-1}^{(t)}, \beta_j^{(*)}, \beta_{j+1}^{(t)}, \ldots, \beta_k^{(t)})^T$
end if
Compute acceptance probability:
$A = \min\left(1, \frac{p(y|\beta^{(*)})}{p(y|\beta^{(t)})}\right)$
Sample $u \sim \text{Uniform}(0,1)$
if $u \leq A$ **then**
    Accept β^* and set $\beta^{(t)} = \beta^*$
else
    Set $\beta^{(t)} = \beta^{(t-1)}$
end if

4 Results

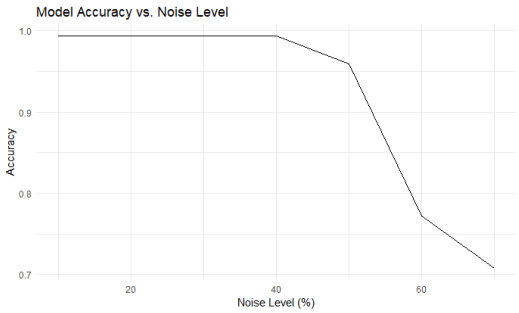
4.1 Results of classification threshold 0.5



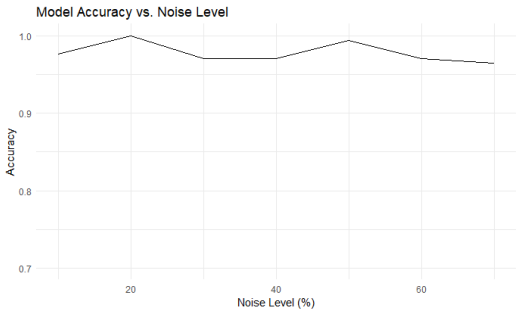
(a) GLM stepwise AIC



(b) GLM stepwise BIC



(c) Metropolis-Hastings MCMC



(d) RJMCMC

Figure 1: Classification threshold 0.5

The results have been grouped together in the above four figures so that they can be referred to for comparison.

As we can see, figure (a) and (b) look almost identical and behave quite similarly. This is quite unexpected as it could be assumed that they are similar in the model selection methods that they employ. Both methods face a gradual decrease in accuracy with increasing noise level. This is expected. However, over 50 iterations of each algorithm, the level of accuracy is at most 0.77 percent. This shows some robustness in the face of noise.

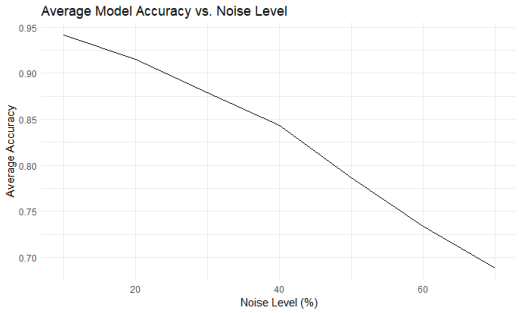
Metropolis-Hastings MCMC performs well until it reaches the 40 percent noise level. After, it drops off steeply.

The best performing algorithm is the RJMCMC algorithm. It is quite robust to noise and there are only minor perturbations as compared to the other three models. The level of noise increase does not seem to affect it at all.

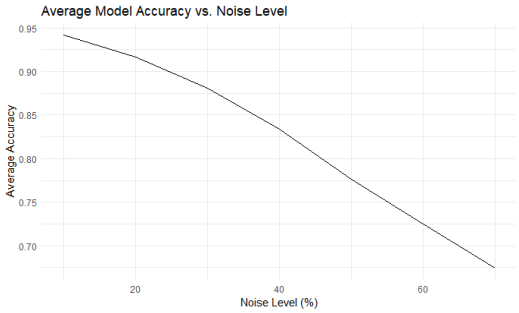
The stark difference is apparent in the different algorithms as we observe a clear trend in Fig. (a), Fig. (b) and Fig. (c) while Fig. (d) does not follow similarly.

What this signals is that there is indeed an effect on the accuracy of the algorithms. In addition, since the results have been iterated 50 times and averaged out anomalies are controlled for as well.

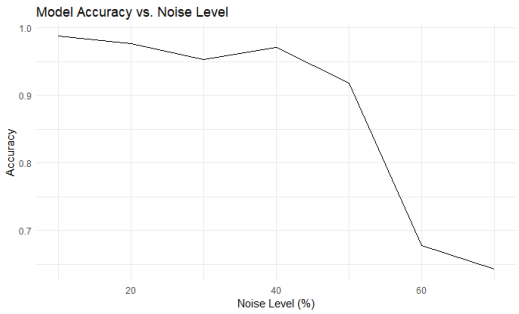
4.2 Results of classification threshold 0.6 to 0.7



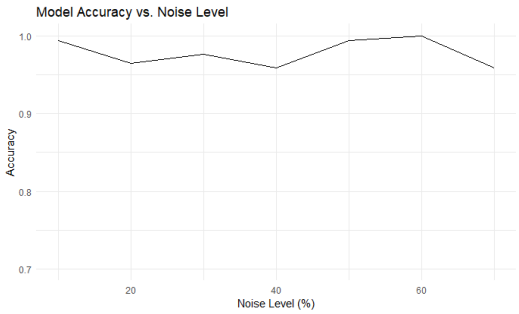
(a) GLM stepwise AIC



(b) GLM stepwise BIC



(c) Metropolis-Hastings MCMC



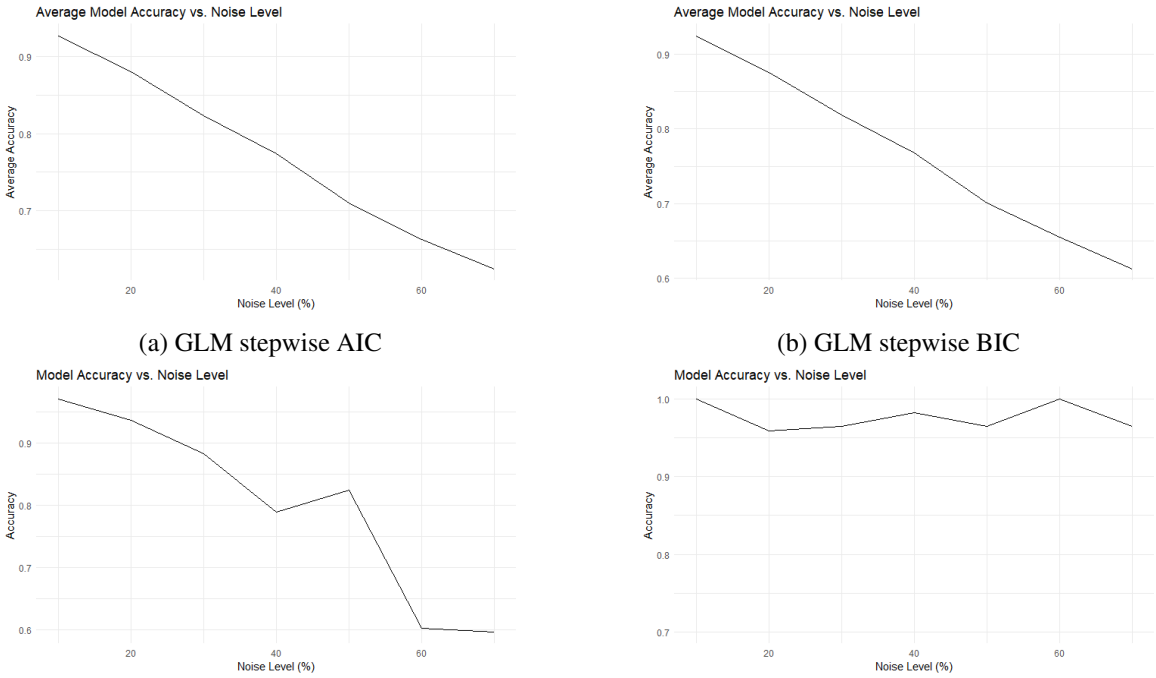
(d) RJMCMC

Figure 2: Classification threshold 0.6

When increasing the classification threshold from 0.5 to 0.6, it can be observed that in Fig. (a) and Fig. (b) that the slopes have been sharper and the decreases in accuracy are slightly increased. The accuracy drops a further 0.05 points, to just below 0.7. The Metropolis-Hastings algorithm seems to perform worse here, though. With quite erratic behaviour in 40 percent noise level and then a sharp decrease just as before. The RJMCMC algorithm performs just as it did before.

The increase in the classification threshold from 0.5 to 0.6 may have an effect on some instances of the data set. For example, those instances which before were right on the precipice of being classified as ”M” (malignant), due to the class noise algorithm had their labels flipped to ”M” now were no longer close to being classified as ”M”. Thus, label flipping has caused them to be a bigger level of noise in the training data.

In Figure 3. we see what happens when the classification threshold increases to 0.7. This time, a much sharper decrease is observed in the slopes of GLM stepwise AIC and GLM stepwise BIC. The trend appears to be the same - however, the sharper decreases indicate that the accuracy drops further at a lower noise level. The accuracy drop in this case reaches a minimum of around 0.6.



(a) GLM stepwise AIC

(b) GLM stepwise BIC

(c) Metropolis-Hastings MCMC

(d) RJMCMC

Figure 3: Classification threshold 0.7

The Metropolis-Hastings MCMC algorithm faces a sharp decrease at the beginning as well, whereas before it was quite accurate until the 40 percent noise level mark. This time it is visible that the accuracy falls 0.2 points until the 40 percent noise level and then decreases sharply again until it falls a bit below 0.6 accuracy.

Again, the RJMCMC algorithm remains as robust as before with barely any decreases in accuracy with increasing noise level. It drops to a minimum of around 0.95

5 Discussion

5.1 Accuracy vs Computational Intensity

In the previous section, four different algorithms were compared, with the RJMCMC algorithm standing out as the most robust against the increasing class noise level. However, viewing this performance in isolation does not paint the full picture; it is essential to consider other factors, such as computational cost, applicability, and ease of implementation, which are often vital in various model selection and inference procedures.

One significant drawback of the RJMCMC algorithm is its computational complexity. When contrasted with the GLM stepwise AIC and BIC procedures, RJMCMC's computational expense becomes clear. The algorithm demands the proposal of an appropriate model, followed by complex calculations to decide on a transition to the proposed model. This ability to "jump" between dimensions requires substantial computational resources, compared to the relatively modest requirements of the optimization and fitting processes in the GLM function, even with the added complexity of stepwise AIC and BIC procedures. In scenarios involving smaller datasets or simpler parameters, employing RJMCMC might be an unwise approach, consuming significant time and resources for a relatively straightforward task.

Furthermore, the RJMCMC approach often requires fine-tuning of function parameters, such as selecting the proposal probability. Even minor variations in these parameters can lead to significant shifts in the outcome. This need for precision contrasts sharply with the GLM method, which is often more straightforward to set up and deploy across a variety of scenarios.

The convergence of RJMCMC may pose additional challenges. Reaching convergence can be time-consuming, and any internal issues can take considerable effort to diagnose and correct. Such complexities are less frequently encountered with GLM methods, which typically provide more user-friendly and easily interpretable solutions.

5.2 Robustness

RJMCMC is flexible in allowing transitions between models with different dimensions and provides an avenue for more accurately fitting the underlying structure of the data. This flexibility is contrasted with other techniques and can accommodate for noise and corruption within data sets.

Due to the Bayesian framework behind RJMCMC, The incorporation of prior information acts as a regularization effect, smoothing out some of the noise and providing resistance against overfitting. This is particularly beneficial when dealing with noisy data, as it ensures that the estimates are less likely to be swayed by random fluctuations.

Class-based noise can easily lead to uncertainties in parameter estimates as it obscures the true underlying relationships. RJMCMC's nature as a Markov Chain Monte Carlo method allows it to sample from the posterior distribution of parameters, capturing the uncertainty and providing a more robust estimation.

RJMCMC also offers avenues for customization and tuning. Through appropriate selection of pro-

posal distributions, priors, and other hyperparameters, the algorithm can be tailored to be more resilient to the noise. Additionally, the ability of RJMCMC to explicitly incorporate a noise model within the data-generating process further enhances its adaptability.

In comparison to RJMCMC, the utilization of GLM with stepwise model selection using AIC is an approach that requires some consideration. Its Frequentist nature relies heavily on observed data, leading to potential sensitivity to noise. Specifically, noise may affect the relationships between predictors and the response variable, resulting in incorrect selection of predictors. This noise can either inflate or deflate the apparent importance of predictors, leading to models that might not accurately reflect the underlying relationship.

On the positive side, the flexibility of the stepwise procedure might make it more robust to model misspecification. Even when the true model includes nonlinear relationships or interactions, stepwise AIC might still identify essential predictors, though it may not capture the true underlying functional form. This adaptability contrasts with its potential vulnerability to outliers. Such outliers as faced in class-based noise might disproportionately influence both the parameter estimates in the GLM and the stepwise selection process. Without proper treatment, this might lead to the inclusion or exclusion of crucial predictors.

GLM stepwise AIC and BIC methods both require significantly less computational power and convergence time than Metropolis-Hastings MCMC and RJMCMC methods. This becomes important when considering the size and dimensions of the data set. On the other hand, more flexible models like RJMCMC might provide greater robustness to noise, especially when the true underlying relationship is not well-captured by a linear model.

In conclusion, the choice between the stepwise AIC approach with GLMs and other complex methods depends on specific research needs and considerations. The popularity of GLMs with stepwise AIC, despite potential vulnerabilities to noise, reflects its usefulness in various situations. Careful handling, including robust regression techniques and validation methods, can enable the method to provide reliable and interpretable results.

6 Conclusion

6.1 Summary of Main Contributions

An analysis of both Frequentist and Bayesian statistical methods was carried out using a noisy data set, followed by a comparison. Moreover, an examination and adjustment of the classification threshold were performed to observe its impact on the model's behavior.

The preliminary exploration involved investigating the influence of class-based noise on a relatively small but extensively studied data set. The methodologies employed by the Frequentist and Bayesian statistical frameworks were closely observed. Through the training and testing of methods derived from both frameworks, insights were obtained regarding the robustness of their approaches and their adaptability to changing classification threshold values.

The relative strengths and weaknesses of these frameworks were examined to allow for a meaningful comparison, aiming to determine if either approach can claim superiority. While definitive conclusions cannot be drawn in this modest study, it has provided insights into the inner workings of these algorithms.

The findings point to the superior robustness of the RJMCMC algorithm to increasing noise levels but also highlighted its computational demands and the need for fine-tuning. In comparison, the GLM procedures offer a more accessible and efficient approach but with potential sensitivity to noise. These insights have implications for model selection, balancing the demands for accuracy, efficiency, and adaptability to noise in various applications.

6.2 Future Work

While the current study focused on specific noise levels and data sets, future work could explore different noise structures, computational optimization strategies, or real-world applications to further understand these trade-offs.

There are many avenues to explore in future studies on this topic. This study only explored one type of noise, which was class-based. There are different types of noise to consider in testing for accuracy. For example, Gaussian noise is a type of statistical noise having a probability density function equal to that of the normal distribution, also known as the Gaussian distribution. It's a common model for general, uncorrelated noise.

Beyond Gaussian noise, other types of noise that may be relevant to future investigations include salt-and-pepper noise, which represents occasional large outliers or errors; speckle noise, often associated with data acquisition errors; and quantization noise, related to the discretization of continuous signals. These various noise types may present different challenges and opportunities in the modeling process, and their exploration could lead to more robust methods.

Other areas of exploration could be comparing methods in different models such as linear regression, Poisson regression, or other regression models. Understanding how different noise types affect various regression models could provide insights into the selection and optimization of techniques for diverse data scenarios. Additionally, considering alternative model selection procedures, optimization

techniques, and parameter tuning strategies might unveil new avenues for improving the accuracy and robustness of statistical inference in the presence of noise.

By embracing this multifaceted approach, future research can delve deeper into the complex interplay between noise, model structure, algorithm behavior, and performance metrics. Such exploration could lead to the development of more adaptable and resilient statistical methods, better suited to handle real-world data complexities.

Bibliography

[1] B. G. A. Dobson J Anette, *An Introduction to Generalized Linear Models*. CRC Press, 3 ed., 2008.

[2] N. van Erp and P. Gelder, "Bayesian logistic regression analysis," pp. 147–154, 2013.

[3] G. Marco, "Project statistical reasoning lecture notes," 2021.

[4] D. Spade, "Markov chain monte carlo methods: Theory and practice," 2020.

[5] Kaggle, "Breast cancer wisconsin (diagnostic) data set," September 2016.

[6] S. Chib and E. Greenberg, "Understanding the metropolis-hastings algorithm," *The American Statistician*, vol. 49, no. 4, pp. 327–335, 1995. Accessed 8 Aug. 2023.

[7] Y. Fan and S. A. Sisson, "Reversible jump markov chain monte carlo," 2010.