Bayesian Logistic Regression for Wine Quality Prediction

Max Li and Lei Meng
Thompson Rivers University

Abstract

Feature selection is a pivotal aspect of machine learning that aims to enhance model performance by eliminating redundant or irrelevant predictors. This project applied Bayesian logistic regression for feature selection, using chemical properties to predict wine quality as a case study. We begin with a traditional logistic regression to establish a baseline, followed by integrating Bayesian methods to address the uncertainties in model parameters. The Bayesian approach utilizes prior knowledge and the Metropolis-Hastings algorithm to estimate posterior distributions, enabling a probabilistic evaluation of feature relevance. Our results reveal the efficacy and challenges of the Bayesian feature selection method, particularly in terms of parameter convergence and the influence of prior settings. Despite some issues with mixing and autocorrelation in specific parameters, the Bayesian model provided a deeper understanding of feature importance, which traditional methods might overlook. Future research should focus on refining prior specifications based on empirical data, implementing more robust validation techniques to ascertain the model's predictive power, and comparing Bayesian feature selection with other machine learning approaches.

1 Introduction

Feature selection has been a critical topic in either regression and classification tasks in machine learning, especially when there are too many features for selecting in a single machine learning task. The process of feature selection can be concluded as steps to remove redundant or non-informative predictors from the model [1]. While from a theoretical perspective, including more features should give more power to classify the response variables into different categories, but practically it is not true in most of the cases. Reunanen [2] summarizes four most critical factors that we need to only include a subset of the variables instead of all. Firstly, it is economic to measure a subset of variables in research. Secondly, by excluding variables that are irrelevant can sometimes improve the accuracy of the model. Thirdly, models will be more time efficient if a limited subset of variables are included. Lastly, knowing what variables to include can gain further insights of the study.

In machine learning, we often use wrapper, embedded, and filter methods for feature selection. Wrapper methods is the process of creating multiple models and evaluate the performance of all models by adding or removing variables to find the most optimal combination. Common wrapper methods include forward feature selection, backward feature elimination, and recursive feature

elimination (RFE). Filter feature selections focus on using statistical methods to explain the relationship between features and the response variable. Some statistical metrics are used to determine what features will be included in the model. Common filter feature selection include correlation, Analysis of variance (ANOVA), and Chi-square test, etc. Filter methods could be much faster compared to Wrapper methods in terms of time complexity, since features that are "less likely" to explain the variation in the response variable will be remove directly. And moreover, filter methods are less prone to over-fitting. Embedded methods combines the advantageous aspects of both Filter and Wrapper methods as feature selection is done by observing each iteration of model training phase. In other words, feature selection and algorithm training are performed parallelly. Embedded methods are generally used to reduce over-fitting by penalizing the coefficients of a model being too large and decay the weight of certain non-informative features. Examples of Embedded methods are Ridge Regression, Least Absolute Shrinkage and Selection Operator (LASSO), and Elastic Net. Each has a different criteria of shrinking coefficients of predictors.

In this project, we are focusing on developing a statistical filter based on Bayesian methods to address the limitation and get a comprehensive understanding of the machine learning model uncertainties. In the initial stages of our analysis, we applied a traditional logistic regression model to predict the quality of wines based on various chemical properties. This method provided us with point estimates for the coefficients of the predictors. In the second stage, we took the uncertainty of the estimated coefficients into account transitioned to a Bayesian logistic regression framework. The general framework is as follows: (i) Incorporate prior knowledge or beliefs about the parameters into the model with prior distributions; (ii) Approximate the full posterior distribution for each parameter through Metropolis-Hastings algorithm; (iii) Excluding or keep predictor variables based on probabilistic criteria.

This project is organized as follows. The next section describes the dataset for performance evaluation of the conventional logistic regression and Bayesian logistic regression, where as section 3 describes the employed Bayesian method and algorithms. Section 4 describes and compares the results and Section 5 concludes the project and discusses directions for further research.

2 Data

Our study employed a dataset compiled from chemical analysis of wine in a specific area of Italy [3]. The dataset contains quality data of both red wine and white wine and we are taking only the red wine data into account in our project. The selected dataset comprises 1,599 red wine samples, each characterized by 11 features representing different tests. The details of the variables are as follows in Table 1.

Data Processing

Initially, the outliers of the dataset were capped using the threshold of 1.5 times the interquartile range (IQR) for each feature to reduce the influence of extreme values that could skew the results.

Then, the quality scores of wines, originally ranging from 0 to 10, were dichotomized into two categories: 'low' (below 6) and 'high' (6 and above) for simplifying the outcome for logistic regression and facilitating a clearer interpretation of model outputs.

Finally the dataset underwent a standardization process. Each predictor variable x_j was

Table 1

Red wine data description.

Name	Role	Type	Description
Fixed Acidity	Predictor	Continuous	Most abundant acids.
Volatile Acidity	Predictor	Continuous	Amount of acetic acid.
Citric Acid	Predictor	Continuous	Citric acid level.
Residual Sugar	Predictor	Continuous	Amount of sugar remaining.
Chlorides	Predictor	Continuous	Amount of salt present in the wine.
Free Sulfur Dioxide	Predictor	Continuous	Free form of sulfur dioxide.
Total Sulfur Dioxide	Predictor	Continuous	Overall concentration of sulfur dioxide.
Density	Predictor	Continuous	Density of the wine
pH	Predictor	Continuous	Level of acidity or basicity of the wine.
Sulphates	Predictor	Continuous	Wine antioxidant additive.
Alcohol	Predictor	Continuous	Percentage of alcohol content in the wine.
Quality	Response	Categorical	Sensory score given by experts.

centered and scaled to have a mean of zero and a standard deviation of one:

$$x'_j = \frac{x_j - \bar{x}_j}{s_j}$$

where $\bar{x}_j$ is the sample mean and s_j is the sample standard deviation for the predictor x_j.

3 Method

Logistic Regression Model

Following data preprocessing, we employ a baseline logistic regression model to associate the dichotomized quality outcomes with the standardized predictor variables. The logistic regression model is formalized as:

$$\Pr(Y_i = \text{"high"} \mid x_i, \beta) = \frac{e^{\theta_i}}{1 + e^{\theta_i}} \tag{1}$$

where Y_i is the binary response variable, in this case we are taking the "high" wine quality into concern. x_i is the vector of predictors for the i-th observation, β represents the vector of regression coefficients. The log-odds of the outcome event θ_i is expressed as a linear combination of the predictors:

$$\theta_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} \tag{2}$$

where β_0 is the intercept and β_j are the coefficients correspond with the predictors.

The model assumes that all available predictors contribute to the estimation of wine quality, serving as a reference for comparison before the process of feature selection.

The coefficients β_j are estimated using maximum likelihood estimation (MLE), where the

likelihood function is given by:

$$L(\beta) = \prod_{i=1}^{n} \Pr(Y_i \mid x_i, \beta)^{Y_i} (1 - \Pr(Y_i \mid x_i, \beta))^{1-Y_i} \tag{3}$$

Logistic Regression with Bayesian Feature Selection

The logistic regression model with Bayesian feature selection framework for the binary response Y_i is similarly given by:

$$\Pr(Y_i = \text{"high"} \mid x_i, \beta, \gamma) = \frac{e^{Z_i}}{1 + e^{Z_i}} \tag{4}$$

The log-odds of the outcome event Z_i is expressed as a linear combination of the predictors:

$$Z_i = \beta_0 + \sum_{j=1}^{p} \beta_j \gamma_j x_{ij} \tag{5}$$

Here, the newly introduced γ_j is a binary indicator variable for the inclusion of the j-th predictor in the model. If $\gamma_j = 1$, the j-th variable is included; if $\gamma_j = 0$, it is excluded.

Prior

Priors express our a priori beliefs about the parameters before observing the data. For the logistic regression model parameters, we specify both the prior for the intercept β_0 and coefficients β_js follow a normal distribution:

$$\beta_0, \beta_j \sim \mathcal{N}(\mu = 0, \sigma^2 = 1) \tag{6}$$

For the coefficient terms β_0 and β_js, a normal prior distribution with $\mu = 0$ reflects no preference for a positive or negative starting point, and the prior variance $\sigma^2 = 1$ implies that large values of the coefficients are less likely a priori.

The binary inclusion indicators γ_j, for each variable are given Bernoulli priors, representing a lack of prior preference for inclusion or exclusion:

$$\gamma_j \sim \text{Bernoulli}(p = 0.5) \tag{7}$$

These priors are chosen to be weakly informative, ensuring that the data have a significant influence on the posterior.

Likelihood Function

The likelihood function is based on the logistic regression model and quantifies the probability of observing the data given the predefined parameters:

$$L(\beta, \gamma \mid Y, X) = \prod_{i=1}^{n} \left(\frac{e^{Z_i}}{1 + e^{Z_i}} \right)^{Y_i} \left(1 - \frac{e^{Z_i}}{1 + e^{Z_i}} \right)^{1-Y_i} \tag{8}$$

Posterior Derivation

The posterior distribution that based on the Bayes' theorem combines the prior and likelihood, providing a probabilistic summary of our updated beliefs about the parameters after observing the data [4]:

$$p(\beta,\gamma \mid Y,X) \propto L(\beta,\gamma \mid Y,X) \times p(\beta) \times p(\gamma) \tag{9}$$

Due to the complexity of the logistic regression likelihood, the posterior distribution cannot be expressed in closed form and thus we must approximate it using Markov chain Monte Carlo (MCMC) computational methods.

Metropolis-Hastings Algorithm

The Metropolis-Hastings algorithm is employed to sample from the posterior distribution when the analytical form is intractable [5]. We first set initial values for the parameters $\beta_0^{(0)}, \beta^{(0)}, \gamma^{(0)}$ and define the proposal distribution variances $\sigma^2_{\beta_0} = 1$ and $\sigma^2_{\beta} = 1$. For each iteration $t = 1, 2, \ldots, T$, the updating process are as follows:

- Update β_0: Propose β_0^* by sampling from $\mathcal{N}(\beta_0^{(t-1)}, \sigma^2_{\beta_0})$; Compute the acceptance ratio α and update $\beta_0^{(t)}$ accordingly.
- Update β_js: Propose β_j^*, conditional on γ_j, from $\mathcal{N}(\beta_j^{(t-1)}, \sigma^2_{\beta}\gamma_j^{(t-1)})$; Compute the acceptance ratio α and update $\beta_j^{(t)}$ accordingly.
- Update γ_j: Propose to switch the state of γ_j to $1 - \gamma_j^{(t-1)}$; Compute the acceptance ratio α and update $\gamma_j^{(t)}$ accordingly.

The acceptance ratio α is defined as

$$\alpha = \min\left(1, \frac{p(\beta_0^*, \beta^*, \gamma^* | Y, X)}{p(\beta_0^{(t-1)}, \beta^{(t-1)}, \gamma^{(t-1)} | Y, X)}\right),$$

where $p(\cdot|Y,X)$ denotes the posterior density.

The convergence of the chain will be assessed after discarding burn-in samples and employ trace plots as diagnostics. We then use the samples obtained from the Metropolis-Hastings algorithm to estimate posterior distributions and make the feature selection based on the posterior simulation result. The complete algorithm is given by this Github link: `https://github.com/maxjinli/DASC5420`

4 Results

Traceplots

The traceplots for β_j and $\beta_j \times \gamma_j$ are shown in Figure 1 and Figure 2. They provide visual diagnostics of the sampling process. For well-mixed chains, we expect to see a "hairy caterpillar" appearance. That represents random fluctuation around a constant value with no systematic trends or patterns.

For $\beta_4, \beta_5, \beta_8, \beta_9, \beta_{10}$, and β_{11}, the traceplots display good mixing, with the chains covering a wide range of values and showing no apparent trends or periodic structures. The traceplots for β_1, β_3, and β_6 also indicate adequate mixing but with slightly more variation, suggesting these parameters may have a more complex posterior landscape. Traceplots for β_0, β_2, and β_7, exhibit less ideal mixing, characterized by longer stretches of high autocorrelation.

For $\beta \times \gamma$ terms, traceplots are interpreted in the context of inclusion probabilities. Generally, the traceplots are fluctuating with no clear pattern, displaying good mixing.

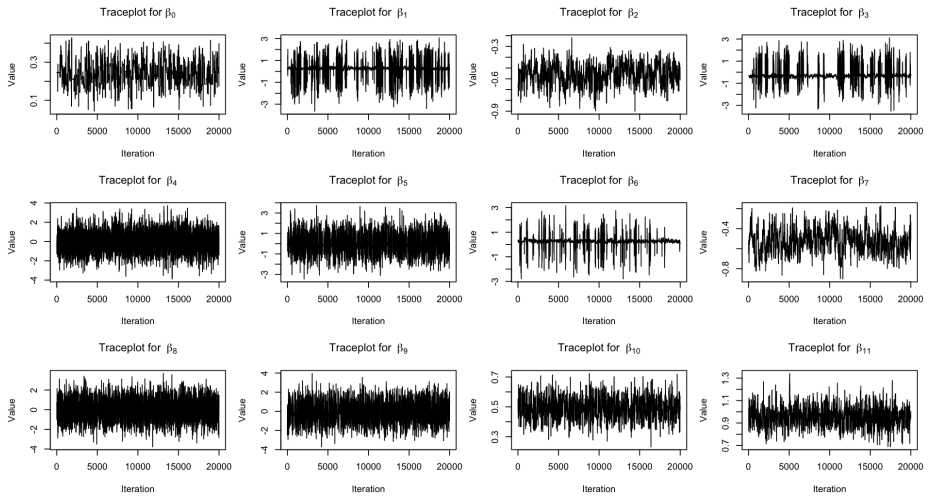

Figure 1

Traceplots for β_j

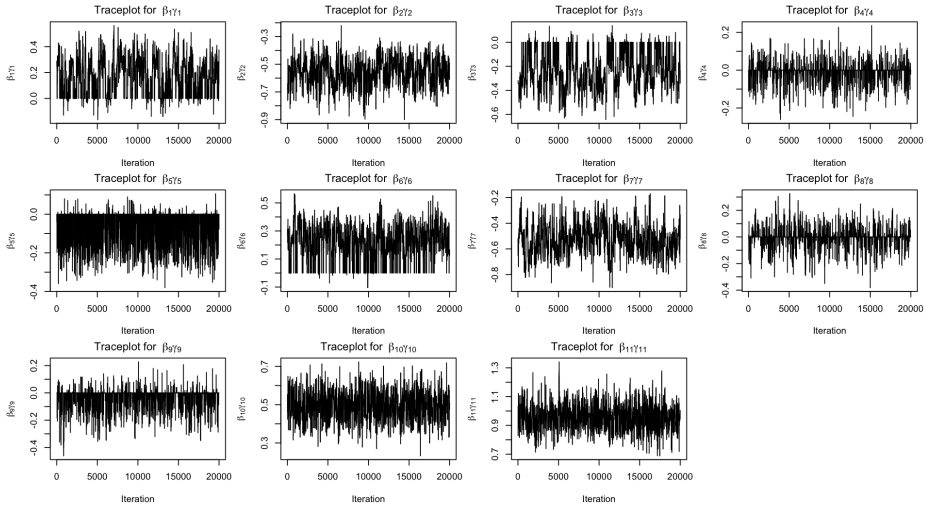

Figure 2

Traceplots for $\beta_j \times \gamma_j$

Effective Sample Size

The effective sample size out of 10,000 for the parameters β and $\beta \times \gamma$ is provided in Table 2 below:

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}
β	2176	319	1384	2274	2581	1763	228	2552	2371	880	784
$\beta \times \gamma$	139	319	106	3682	1118	241	228	776	451	880	784

Table 2

Effective Sample Size out of 20000

The effective sample size for the parameters reflects the number of independent samples that are equivalent to the correlated samples obtained through MCMC. Higher effective sample size values are indicative of better mixing and less autocorrelation within the chain.

For β, parameter 2 and 7 show samples considerably less than half of the total iterations, suggesting poor mixing and high autocorrelation. The rest exhibit higher samples, indicating a better mixing of the chain for these parameters. For $\beta \times \gamma$, the samples for parameter 1, 2, 6, 8 and 9 decreases when considering its interaction with γ, which may indicate a loss in efficiency when this parameter is included in the model. The samples for parameter 4 increases significantly, suggesting that the inclusion of this parameter in the model leads to better mixing and thus more informative sampling. Parameters 2, 7, 10, and 11 have similar samples for their interactions with γ, implying consistent levels of autocorrelation and chain mixing whether the parameter is included in the model or not.

Posterior Probability

The posterior probability of the top five most frequently occurring values of γ was approximated, and the results obtained are in Table 3. The density plots for $\beta \times \gamma$ are provided below in Figure 3 illustrating the distribution of the MCMC samples for the product of the regression coefficients and their respective inclusion indicators.

γ Values	Posterior Probability
11100110011	0.20275
11101110011	0.13860
01000110011	0.10395
01001110011	0.05115
01100110111	0.04140

Table 3

Top five most frequently occurring values of γ and their posterior probabilities

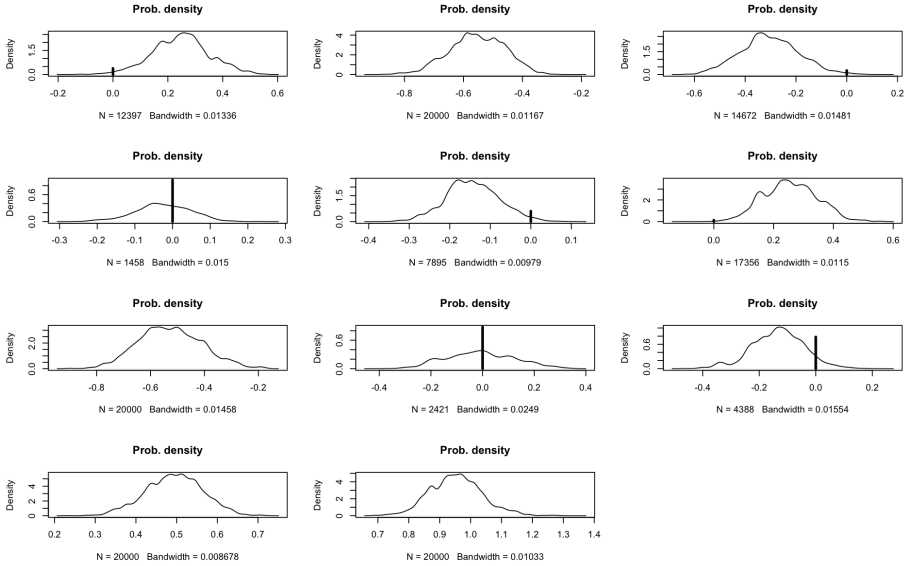

Figure 3

Traceplots for $\beta_j \times \gamma_j$

5 Discussion

In this project, we successfully implemented a Bayesian logistic regression framework to perform feature selection in predicting wine quality based on chemical properties. Our approach highlighted the role of priors and the use of the Metropolis-Hastings algorithm for posterior estimation. The traceplots and effective sample sizes showed varying degrees of mixing across parameters, suggesting that while some features were well-represented in the model (like β_4 and β_5), others (such as β_2 and β_7) experienced poor mixing, indicating potential issues in parameter estimation or model fit.

Moreover, the posterior probabilities for the inclusion indicators (γ) provided a probabilistic assessment of each feature's relevance, offering a data-driven way to understand feature importance which is less prone to overfitting and more informative than traditional methods.

While our current study explored a foundational framework for Bayesian feature selection in predicting wine quality, there remain several opportunities for enhancement and further research. The points below outline potential research directions that could address some of the limitations identified in this study and expand the utility of the Bayesian approach.

- Future investigations could focus on refining prior specifications, using empirical data to inform these choices or employing hierarchical structures to manage feature complexity.
- More rigorous validation frameworks, including cross-validation and external validations, should be implemented to comprehensively understand the model's predictive accuracy and its applicability to various types of similar data.
- Comparative studies should be assessed to evaluate the relative effectiveness of Bayesian feature selection against other machine learning approaches.

References

[1] Max Kuhn, Kjell Johnson, et al. *Applied predictive modeling*. Vol. 26. 488. Springer, 2013.

[2] Juha Reunanen. "Overfitting in making comparisons between variable selection methods." In: *Journal of Machine Learning Research* 3.Mar (2003), pp. 1371–1382.

[3] Stefan Aeberhard and M. Forina. *Wine*. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5PC7J. 1991.

[4] Peter D Hoff. *A first course in Bayesian statistical methods*. Vol. 580. Springer, 2009.

[5] Edward I George and Robert E McCulloch. "Variable selection via Gibbs sampling." In: *Journal of the American Statistical Association* 88.423 (1993), pp. 881–889.