Writing Evaluation Score Prediction using Bayesian Approaches

Andrew M Henrichsen, Sehee Park, Yao Zhao
Applied Mathematics and Statistics
Johns Hopkins University

January 22, 2023

Abstract

There are automated tools that help with improving written English, but there does not exist an evaluation tool that can generate comprehensive feedback on the writing's quality. Aiming to fill this blank area, we develop a method to automatically grade vocabulary proficiency in a practice English essay using Bayesian analysis. Particularly, we proposed a 'Bagged Mask Importance' to handle the sparsity with Horseshoe prior, which can possibly be extend to general Natural Language Processing tasks.

1 Introduction

Writing practice is an efficient approach for students to learn English. However, writing assessment requires subjective evaluation from teachers, which is time-consuming and limits the frequency of practice. Building a model to automatically evaluate and grade essays written by people learning English will greatly reduce the workload of English teachers and provide an opportunity for English learners to practice as much as they want. Currently, the primary focus of the project is the prediction of vocabulary scores using different Bayesian approaches and the interpretation of the models. We start from the classic Naive Bayes model and shift to a General Linear Model with shrinkage priors to handle the sparsity. To handle the problem of high dimensionality, which is common for Natural Language Processing tasks, we use word bagging on a masked word importance feature. The simulation results and accuracy test shows the Bayesian linear model does provide higher accuracy and model robustness compared to several classical Machine Learning algorithms.

2 Literature Review

Though Natural Language Processing studies have started since 1950s, Bayesian methods were not broadly applied until 1980s and the early Bayesian approaches

for NLP was mostly based on Bayesian point maximization, where the prior serves the role of a penalty term. After 2000, Bayesian methods were more widely utilized, such as applications of the Bayesian hidden Markov models [4][5]. Currently, Bayesian methods are broadly used in modern Natural Language Processing tasks including machine translation, information retrieval, text summarizing, question answering, information extraction, topic modeling, and opinion mining.[8] Also, with the development of computer science and increasingly available computing resources, the use of deep learning models such as Recurrent Neural Network and its variants Long Short-Term Memory[5], Transformers [6] and BERT[7] have become practical. These models have been broadly used in both academic research and industrial application, including for evaluating writing. However, though deep learning models provide great accuracy for writing evaluation tasks, the huge amount of parameters inevitably leads to the related problems of high computational complexity and poor human interpretability.

3 Proposed Method

3.1 Naive Bayes Model

Naive Bayes Model is broadly used for text classification tasks. As a relatively concise model with fast computation process, it is a good starting point and benchmark for comparing different methods. The Naive Bayes model has a posterior predictor as following:

$$Y_i = arg\max_{y_i \in Y} \hat{P}(y_i|d) = arg\max_{y_i \in Y} \hat{P}(y_i) \prod_{1<k<n} \hat{P}(x_k|y_i) \tag{1}$$

with an empirical prior estimated from data:

$$\hat{P}(y_i) = \frac{N_{y_i}}{N_{Total}} \tag{2}$$

where Y_i is the random variable, y_i is the numbers of classes or scores for prediction, n is the dimension for features, in this task, dimension of dictionary, and N is the number counting functions. The Naive Bayes model takes individual words as input features to calculate their likelihood under different classes, and the sum of likelihood of all features is the probability of belonging to different classes. As mentioned above, the model has advantages of fast computation and providing good performance even with small size sample data. What's more, Naive Bayes model handles multinomial classification tasks well, since likelihoods are calculated independently for each class, and it is not sensitive to missing data. On the other hand, Naive Bayes model makes a strong assumption about the independence between features. Consequently, if the features are high dimensional and there is high correlation between features, then Naive Bayes will have low accuracy. Additionally, because the prior for Naive Bayes is actually for the probability distribution for each class estimated from the sample data, the prior doesn't carry any information and knowledge from linguistic perspective.

3.2 General Linear Model

Considering the drawbacks of the Naive Bayes model, we propose a Bayesian general linear model (GLM), which uses a Bayesian linear regression framework. The model is defined as:

$$Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 X_{3i} + \beta_4 X_{4i} + \beta_5 X_{5i} + \epsilon_i \tag{3}$$

where X_{1i} to X_{5i} are five features extracted from the sample data i, ϵ is the residual, β_0 to β_5 are parameters need posterior inferences. We construct these features based on linguistic knowledge about how to evaluate vocabulary knowledge. $X_1 \in \mathbb{R}^n$ is Bagged Mask Exposure to Importance; $X_2 \in \mathbb{R}^1$ is the average sentence length of the text, which is explained later in Section 4.1.2; $X_3 \in \mathbb{R}^1$ is the average length of the top ten longest words in the text; $X_4 \in \mathbb{N}^1$ is the number of unique words that appear in the text; and $X_5 \in \mathbb{N}^1$ is the number of spelling errors. Note that the combination of the variables has high dimension, which is understandable because text classification tasks generally have high dimensional parameter spaces. For the purpose of better interpretation and reducing the dimension of parameter space, we applied horseshoe prior[8,9,10] to deal with the sparsity. If we stack all the features into one single variable X, the model can be expressed as:

$$Y_i = \beta X_i + \epsilon_i, \text{ where } \epsilon_i \sim N(0, \sigma^2) \tag{4}$$

and the horseshoe prior is:

$$\beta_j | \lambda_j \sim N(0, \tau^2 \lambda_j^2),\ \lambda_j \sim C^+(0, 1) \tag{5}$$

Hyper-parameters τ and λ denote the global and local shrinkage parameters, which control the posterior shrinkage of parameters. As for β_0, we set a normal prior with mean at 0 and variance of 1 based on our model output interval. In conclusion, GLM provides a possible approach to do posterior inference despite the sparsity of data. Also, with the posterior estimations of parameters, it is easy to analyze the contributions from each feature to the model. Thus, we are able to reconstruct a scoring decomposition and give reasonable feedback for sample texts.

4 Simulation

4.1 Data Processing

4.1.1 Tokenization and Lemmatization

The data set is from Kaggle, containing 3911 sample essays from eighth to twelfth grade students, scored on 6 metrics: cohesion, syntax, vocabulary, phraseology, grammar, and conventions. The articles and their corresponding vocabulary scores were used as raw data. One standard approach in NLP task is to tokenize the text by breaking texts into sequence of individual words. The package

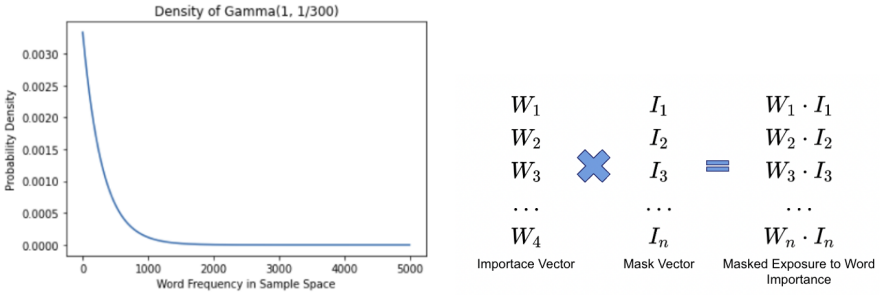


Figure 1: Gamma density and masked word importance

Natural Language Toolkit (NLTK) was used to perform the tokenization. After removing numbers and punctuation there are approximately 21000 unique words in the sample data. Since the task aims at evaluating vocabulary level of the text, some words with different spelling have exactly the same contribution to the model such as verbs in different tenses. Thus, we used lemmatization, which only use one word to represent the linguistic family. Lemmatization was an effective approach to reduce dimensions, bringing it down to about 7000. Then, a dictionary was constructed to record the words and their features such as length and frequency.

4.1.2 Bagged Mask Importance

Based on the study of Dr. Zipf in 1949 [11], the rank of word usage frequency follows a Zipf's distribution with a probability mass function of:

$$P(k) = \frac{1/k^s}{H_{N,s}}, \text{ where } s \geq 0, N \in \mathbb{N}^+, H_{N,s} \text{ is the Nth harmonic number} \quad (6)$$

This probability mass function gives a huge tail, indicating that a large proportion of words would have very low usage frequency. Though low frequency words are universal in NLP tasks, they cannot be disregarded, because they carry extra information. To handle this long sparse tail, we designed a feature named Masked Exposure to Word Importance (Figure 1). The importance of $word_j$, denoted by W_j, is defined as $W_j = f(q_j)$, where q_j is the frequency of $word_j$ in the sample space, f is the probability density function of $Gamma(1, \frac{1}{300})$(Figure 1). The gamma density was designed to compress the importance of words with high frequency. From the plot, the importance of word with frequency in the whole corpus higher than 2000 are squeezed towards 0. The mask vector is composed by indicator functions showing if $word_j$ is used in a specific sample text. By applying pairwise multiplication of the importance vector and the mask vector, the masked exposure to word importance is a vector carrying the importance of words used in a certain text. Introduction of such a feature maps the initial frequency to a new importance distribution and values the low-frequency words with high importance so it is possible to adapt traditional Bayesian methods

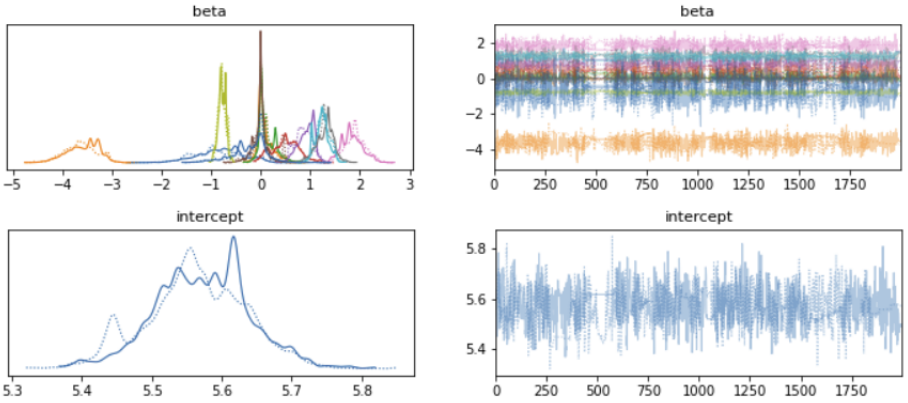
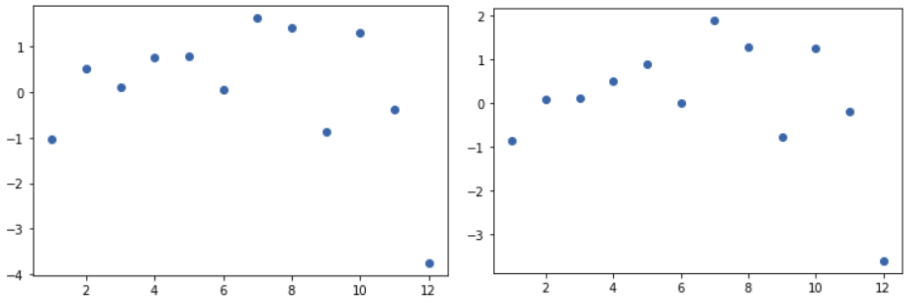


Figure 2: Posterior KDE and trace plots of beta and intercept

such as Horseshoe prior.
However, the masked importance feature has a dimension equal to the dimension of dictionary constructed from sample data, which is 7569 in our model. After flattening with X_2 to X_5, there are 7573 parameters (excluding the interception) that needs to be inferred. With several experiments, it is hard for MCMC simulations to converge to posterior distribution within an acceptable time, and it is also unnecessary to estimate parameters for every single word in the dictionary. Thus, we applied another common NLP technique of word bagging, by dividing the importance into a partition of subsets and assigning numbers for different bags. Then the sums of word importance belong in each bags are used as exposure to bagged mask importance. We used 8 bags of labels 1 to 8, with increasing importance levels, leading to a total parameter number of 12 (excluding interception).
Notice that the bagged mask importance has exposures that are much smaller than other features, thus, a batch normalization is required for direct comparison between posterior estimations of the parameters. Since 0 has a non-existence nature in our features, we used Min-Max normalization so that there will not be any negative exposures and able keep 0s at the same time.

4.2 Markov Chain Monte Carlo Simulation

We used PyMC3 package in python to conduct MCMC simulations. NUTS sampler[14] was applied with 1000 iterations as the burn-in stage and 2000 additional iterations as samples from posterior distributions. Two Markov chains were sampled individually and the sampler gave an acceptance rate of 0.7. The kernel density estimations for posterior distributions are shown in Figure 2. We used the average of two chains to estimate the posterior value of β.



(a) MLE estimator (b) Posterior estimator

Figure 3: Gamma density and masked word importance

5 Results and Conclusion

5.1 Parameter Estimation Analysis

First, we would like to take a look at the estimation of parameters via different approaches. From figure 3, we can tell that the parameters estimated by MLE and by Bayesian GLM posterior estimation are generally similar to one another, except the GLM estimators shrank by different values. By checking the exact values of estimators in Table 1, β_2 shrank towards zero due to the horseshoe prior we applied on GLM. Though there is no significance amount of shrinkage occurred in our model, it is testified to be applicable through constructing the masked importance feature.

We also attempted to apply GLM with horseshoe prior on unbagged mask exposure to importance, but it was time consuming to run the MCMC simulation. The chain doesn't converge after a burn-in stage of 10000 iterations, which takes more than 10 hours to process. However, the MLE of β (Figure 4) shows that there are a lot of parameters that make trivial contributions to the score prediction. There are 4257 out of 7573 parameters that have values less than 0.01, and it is meaningful to explore possible solutions with the high dimensional case.

Table 1: Estimations of β

	β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8	β_9	β_{10}	β_{11}	β_{12}
MLE	-1.0442	0.5166	0.1001	0.7507	0.7974	0.0611	1.6234	1.4106	-0.8609	1.3130	-0.3906	-3.7455
GLM	-0.8487	0.0091	0.1075	0.4875	0.8844	-0.0005	1.8797	1.2858	-0.8346	1.2563	-0.1831	-3.6057

From Table 1, β_7 and β_8, corresponding to the seventh and eighth word bags, has the highest values. This result is in accordance with our previous assumption that words with high importance would contribute more to the predicted scores. What's more, β_1 and β_{12} has the lowest values, which are negative. It is also reasonable to believe an article with more low level words and spelling error would have lower scores. Surprisingly, β_9 also have a significant negative value,

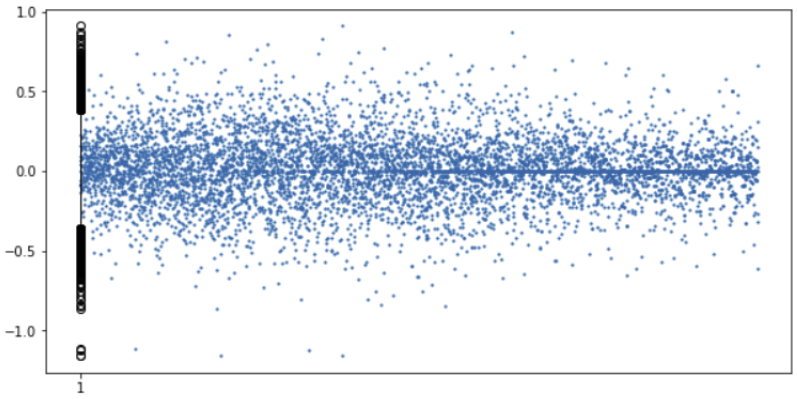


Figure 4: MLE of unbagged mask importance parameters

which is a contradiction to our prior knowledge. It implies that texts with longer average sentence length would have lower vocabulary scores. A possible explanation would be that students with high vocabulary scores tends to use advanced words and write more precise sentences. Another possibility is that this was strongly influenced by a few outliers where students used no or few periods in their essays. Though this would be a grammar error instead of a vocab error, poor grammar is often correlated with poor vocabulary.

5.2 Model Comparison

For evaluation the model's accuracy, we used the Average Squared Error defined as following:

$$ASE = \frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}, \text{ n: the number of samples evaluated} \tag{7}$$

The ASE allows a direct comparison on samples sets with different sizes. We split the data in a 0.6/0.4 proportion of training/test sets and trained Naive Bayes, SVM, OLS, GLM for the purpose of comparison (Table 2). Compared

Table 2: ASE of differnt models

	Naive Bayes	SVM	OLS(no bagging)	OLS(with bagging)	GLM(with bagging)
Training ASE	0.2230	0.5827	0.000	1.0574	1.0379
Testing ASE	1.2901	1.1527	2.5016	1.0658	1.0447

to other popular Machine Learning methods, proposed GLM model has the lowest ASE. Compared to OLS methods, GLM with horseshoe prior reduced one parameter (β_2) and slightly increases model accuracy, which testifies the efficiency of horseshoe priors. What's more, the ASE of OLS method without bagging shows that the high dimension of parameters causes over-fitting problems,

which means adding regularization terms such as LASSO or using priors for the purpose of regularization is necessary for our task. Additionally, the results for OLS with bagging and GLM tells that word bagging effectively increases the accuracy as well as the robustness of model.

5.3 Conclusion

The Bayesian General Linear Model provides a better accuracy with more robust results compared to other models. However, the most strong support of using Bayesian approaches is that the model gives a more easily interpretable posterior distribution and helps with understanding the linguistic properties. From the results, the model not only demonstrates the effectiveness of using word importance, but also tells us some linguistic characteristics about vocabulary efficiency evaluation. However, there are two major drawbacks of the current model. First is the convergence of MCMC with huge parameter space as mentioned above. What's more, since we are only use positive numbers to represent the "presence of features", context and grammar are not included into the model, which implies one can trick the model by using nonsense high importance vocabulary words to obtain a high score. In this case, a mixed model combining interactions between individual predictions of all six grading categories would be a better way to ultimately achieve the goal of writing evaluation.

References

[1] Alexander, J.A. & Mozer, M.C. (1995) Template-based algorithms for connectionist rule extraction. In G. Tesauro, D.S. Touretzky and T.K. Leen (eds.), *Advances in Neural Information Processing Systems 7*, pp. 609–616. Cambridge, MA: MIT Press.

[2] Bower, J.M. & Beeman, D. (1995) *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System.* New York: TELOS/Springer–Verlag.

[3] Hasselmo, M.E., Schnell, E. & Barkai, E. (1995) Dynamics of learning and recall at excitatory recurrent synapses and cholinergic modulation in rat hippocampal region CA3. *Journal of Neuroscience* **15**(7):5249-5262.

[4] Goldwater, S., Griffiths, T. L., Johnson, M. (2007) Bayesian framework for word segmentation: Exploring the effects of context. Cognition, 112, 21–5

[5] Sepp Hochreiter and Jürgen Schmidhuber. (1997) Long Short-Term memory. Neural computation, 9(8):1735–1780

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. (2017) Attention is all you need. CoRR, vol. abs/1706.03762

[7]Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K.(2018) Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805

[8]Carvalho, C. M., Polson, N. G., Scott, J. G. (2009, April). Handling sparsity via the horseshoe. In Artificial Intelligence and Statistics (pp. 73-80). PMLR.

[9]Piironen, J., Vehtari, A. (2017). Sparsity information and regularization in the horseshoe and other shrinkage priors. Electronic Journal of Statistics, 11(2), 5018-5051.

[10]Carvalho, C. M., Polson, N. G., Scott, J. G. (2010). The horseshoe estimator for sparse signals. Biometrika, 97(2), 465-480.

[11]Zipf GK (1949). Human Behavior and the Principle of Least Effort. Cambridge, Massachusetts: Addison-Wesley. p. 1.

[12]Jen-Tzung Chien. 2019. Deep Bayesian Natural Language Processing. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts, pages 25–30, Florence, Italy. Association for Computational Linguistics.

[13] Ueffing, N., Ney, H. (2004, October). Bayes decision rules and confidence measures for statistical machine translation. In International Conference on Natural Language Processing (in Spain) (pp. 70-81). Springer, Berlin, Heidelberg.

[14]Hoffman, Matthew D., Gelman, Andrew. (2011). The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo.