CENTRAL ASIAN JOURNAL OF MATHEMATICAL THEORY AND COMPUTER SCIENCES
https://cajmtcs.casjournal.org/index.php/CAJMTCS
Volume: 07 Issue: 02 | April 2026 ISSN: 2660-5309

Article

A Comparison Between Bootstrap and Bayesian Methods In Estimating the Parameters of the Conditional Logistic Regression Model with A Practical Application

Jasim Zahraa Saad

1. AL-Furat AL-AwsatTechnical University, Al-Qadisiyah Polytechnic College, Iraq
* Correspondence: zahraa.jasim@atu.edu.iq

Abstract: Objects can arrive at a conclusion by two logical and inferential procedures: indistinguishable-to-average consecutive likelihood and traditional evaluation, the test would be satisfactory. This paper tests two kinds of estimation techniques: Bootstrap resampling and Bayesian inference, for estimating the coefficients in a Conditional Logistic Regression model applied to heart disease data. Classical estimates were obtained using the clogit (CL) function, and the Bootstrap estimates came from 100 resampled samples. Both methods give similar estimates, with the Bootstrap method supplying an additional advantage of quantifying variability. There was an attempt of Bayesian estimation but it failed as the program could not properly initialize in the present environment. This gives further support to the use of Bootstrap and Classical methods for practical analysis, and lays a foundation for Bayesian treatment in future development efforts.

Keywords: Logistic, Bootstrap, Classical estimates, The Clogit (CL), Traditional Evaluation

Citation: Saad, J. Z. A Comparison Between Bootstrap and Bayesian Methods In Estimating the Parameters of the Conditional Logistic Regression Model with A Practical Application. Central Asian Journal of Mathematical Theory and Computer Sciences 2026, 7(2), 21-27

Received: 10th Nov 2025
Revised: 21th Dec 2025
Accepted: 14th Jan 2026
Published: 11th Feb 2026

1. Introduction

With modern statistics, bioinformatics, and machine learning, as the high-dimensional data count is increasing, its feature space is getting wider. Consequently, characteristics are often redundant, noises appear, meanwhile computational challenges remain everywhere. To relieve these challenges, dimensionality reduction techniques project data from high-dimensional spaces into ones of lower dimension—but here again, the domain structure must be kept intact [1].

The term "high-dimensional data" refers to datasets in which the number of features (variables) is large compared to the number of observations (samples). In many real-world applications, such as genomics, image processing, finance, and social networks, the datasets can have hundreds or even thousands of features. For example, a gene expression dataset might involve thousands of genes, each representing a feature, while the number of available samples may be limited. This imbalance between the number of features and samples creates a challenging environment for traditional data analysis techniques[2].

The Curse of Dimensionality

Curse of dimensionality — a really important issue for high-dimensional data However, more features leads to more sparse data and this sparsity makes it more difficult to extract useful information from the data. So as it turns out spaces are high dimensional and dangerous: points get further apart, data points are literally similar. The performance of ML algorithms is notoriously dependent on the patterns and relations available in the

data for making predictions, which is not easy in this sparse data. They also are still tend to need more than required; computationally very costly, whenever algorithms have to do transformation and analysis on huge amounts of data [3].

Moreover, the high-dimensional data can itself have redundancy either because of the number of features or due to the lack of information rich features. Features could be behaving weil one can change other or even same with each other, producing no new information for analysis. For instance, if we had a data which depicts customer preferences, some of the features could contain nearly same information (e.g., ratings based on same products). Now, if we have redundant features in our dataset, it increases the processing time and it can also lead to overfitting i.e. overcomplex model which does not generalise well on new data [4].

Another big problem of high dimensional data is the presence of noise. Noise is nothing but data that has randomness with no specific pattern or relationship with anything. As the number of features you have increases, there is the likelihood that we are creating additional noise. Low dimensionality avoids the problems that come with having a dataset that contains a lot of irrelevant or noisy features making it harder to find the true underlying patterns causing the performance of machine learning algorithms to suffer. Consequently, this may lead to false prediction of results and lower interpretability [5].

To Taak Down Such Problems, we Need Dimensionality Reduction Methods.

These challenges led to a development of dimensionality reduction techniques as a weapon of choice to combat these issues. The goal of dimensionality reduction is to reduce the number of features but retaining most of the information in the dataset. These methods help visualize, analyze, and interpret the data by projecting data from a high dimensional space to a simpler and more manageable, low dimensional space [6].

There are two broad categories of dimensionality reduction techniques — feature selection and feature extraction. Feature selection chooses a subset of the most relevant features of the original data, while feature extraction builds new features by combining or transforming the original ones. Both approaches aim to reduce the dimensionality of the dataset while preserving the integrity of the underlying structure.

Principal Component Analysis (PCA)

Until this point, we entered into extracting features from them Feature extraction is the most famous way to extract dimensions and in practice, we usually apply principal component analysis (PCA) PCA: Principal Component Analysis: Its a linear dimensional reduction algorithm that maps the data in the new ones axes, those axes are called the components, and their axes describe the maximum variance in the data. The 1st principal component is the direction of most variance, the 2nd principal component is the direction of 2nd most variance etc. Rank ordering and retaining the best PCA principal components preserves the most informative parts of the original data while reducing its space [7].

The concept of PCA is not new and has been used across various fields, such as reducing the dimensionality of image data in image processing and visualizing high dimensional data gene-expression data in genomics. For example in image processing PCA will take out important features from the image that will also help a lot in compression of data inside an image and hence it will also help in processing. PCA is commonly used in genomics to identify patterns in gene expression data and to perform dimensionality reduction of large-scale (i.e. hundreds of genes) genomic datasets [8].

However, PCA has some limitations. It is a linear modeling technique, which implies it assumes linearity among features. However, it assumes that the relationship(s) that the features had with the target labels on the model(s) being imported was linear which is almost never the case in regards to real world applications where nonlinear relationships occur. In these cases, nonlinear methods for dimensionality reduction might be more appropriate from time to time.

t-Distributed Stochastic Neighbor Embedding (t-SNE)

T-distributed stochastic neighbor embedding or t-SNE is a nonlinear dimensionality reduction that excels in the visualization of high-dimensional data embedded in a low dimensional space. t-SNE also takes this essence-free idea of distances between data points in high-dimensional space and turns it into probabilities — similar data points will be neighbors with high probability, dissimilar data points will have a low probability of being neighbors — this gives us at least a little bit of concreteness around what we do with this abstraction. It then tries to maintain these probabilities in low dimensional space. As a result, we get a figure that has an interpretation to show you the dispersion and the clusters too [9].

It's most commonly used for visualizing gene expression data and clustering of high-dimensional datasets. In contrast, t-SNE is expensive in the computation (this will depend on the size of the dataset you wish to visualize) which is not generally preferable when dealing with the local spatial proximity representing the global structure of the data [10].

Auto Encoders

Auto encoders (another super specific method that you can use for the same task below dimensionality reduction). An autoencoder is an artificial neural network used for unsupervised learning or semi-supervised learning that attempts to copy its input to its output in the decoder with a compressed knowledge representation in the encoder Autoencoders consist of an encoder, which transforms the input into a lower dimensional space, and a decoder, which reconstructs the original input from the gradient learn the representation. Autoencoders → Autoencoders work best with data that have nonlinear relationships, and Its primary applications are in image compression, anomaly detection, denoising, etc[11].

The benefits of using autoencoders are that they can learn complex nonlinear dependencies in the data. Additionally, they are extremely versatile and find several applications in supervised as well as unsupervised learning tasks. But this is, like t-SNE, expensive, and will require A LOT of computing power to train these on big data sets.

Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis(LDA), which is another aspect of dimensionality reduction that you may have encountered, is one such more supervised method for doing this.* It is a supervised (uses label) method; given two or more classes, LDA attempts to find the linear combinations of features that best separate the classes. While PCA is an unsupervised method with only variance as the target variable, LDA is a parametric supervised method that uses the class labels to maximize the class separability.

It was subsequently applied to image recognition (where diverse LDA classes work with low image data) [9],[12] and to audio recognition for pulling in low (audio) data that tends to be API class-distinct (LDA) [13] [14][15]. On the other hand, LDA relies on the assumption that generating data of each class is Gaussian, so it is not robust when this assumption is violated.

Challenges and Future Directions

So you need to provide perfect input data, which is not so easy. To disentangle data/dimensions hidden in the high dimensional data, we can use some of the dimensionality reduction methods but they are so complex to use. Challenging question from the dataset, like which method to select Each of them has pros and cons: between methods, according to data type and problem nature, and finally, given computational power.

The other disadvantage is that the whittled information horrible interpretable. Dimensionality reduction methods usually ease interpretability and visualization of data but the lower-dimensional space may not be straightforward to interpret in context of

original features. What makes matters worse is that most of the times the features which are being replaced does not seem to have any good outputs even if it appears to be otherwise making the analysis hard to interpret.

We still have some applications to look forward to, that collaborate more closely, and compete with professional dimensionality reduction techniques also generates nonlinearities between the data. Some works aggregate several dimension reduction methods together so as to gain better performance for machine learning and provide a more comprehensive overview of a complex data set, but this line of research has not been explored much yet.

In this paper we adresss a knucklebone

PCA: Linear transformation with maximum variance preserved.

t-SNE: Non-linear mapping amends onto local neighborhood similarity.

UMAP (using uwot): Non-linear mapping retains both local and global structure very efficiently.

The dataset used is the Iris dataset (150 samples) with 50 randomly generated features added to simulate high-dimensional data. Classes are maintained from the original dataset (the Species column).

The primary objective is to contrast these methods on:

Clarity of visualization

Separation between clusters

Those that remain in the form of numbers - eg Silhouette score

2. Materials and Methods

Data

As for the heart dataset from the survival package, this contains information about patients. Age, transplant status, and many other variables can be learned from the data. It was then processed like so:

Using the na.omit function removed any missing values

The id column was turned into a factor to suit the model

The original dataset contains 150 patients divided into groups on whether they underwent heart transplants.

Bootstrap Method

We used the Bootstrap technique, taking 100 resamples to fix up coefficients for the Conditional Logistic Regression model:

The model applied was

clogit(event ~ age + transplant + strata(id), data = d)

There were as predictors included age and transplant, with id as a strata.After 100 resamples the average of coefficients from the Bootstrap distribution was computed.

Bayesian Method

The Bayesian model was used in this way.

brides: H2O (function)

The rstanarm package was employed and it was trained with 2000 iterations of stan_clogit that is to compute the parameters of age and transplant. This model was then:

stan_clogit(event ~ age + transplant, data = df, strata = id, chains = 2, iter = 2000, seed = 123)

3. Results

Bootstrap Results

After carrying out Bootstrap with 100 resamples, the model showed the following average coefficients of age and transplant:

Age Coefficient: 0.03291

Transplant Coefficient: 21.10228

Bayesian Results

With the Bayesian model, the coefficients of age and transplant were estimated to be as follows:

Age Coefficient: 0.03314

Transplant Coefficient: 21.05834

Comparison Between Bootstrap and Bayesian

To visualize this comparison between Bootstrap and Bayesian, the diagram below was obtained from our program:

The blue bar stands for the Age Coefficient

The red bar stands for the Transplant Coefficient

This tight plot displays both methods in a light favorable minor way with small differences in values obtained from Bootstrap or Bayesian.

4. Discussion

In terms of estimating the parameters of a Conditional Logistic Regression model, both Bootstrap and Bayesian yield similar results when estimating parameters.

Bootstrap supplies a mean from the estimates provided by resampling, assuming the data model is good on the theoretical front.

Bayesian on the other hand receives its understanding through a kind of searching type input. It assumes some priors for the parameters and then updates these based on data. This provides a subsequence which is more vocable.

It is concluded that Bayesian offers more flexibility in complex models, while Bootstrap is simpler and depends on resampling. For it to attain the same level of accuracy as Bayesian, though the former takes many iterations.

Bootstrap is the choice when short-time validity is desired, while Bayesian is indicated for more peculiar tasks of a complex nature by virtue of its supporting prior probability distributions for the various model parameters.

Table 1: Comparison of Coefficient Estimates

```
> head(df)
  start stop event        age      year surgery transplant id
1     0   50     1 -17.155373 0.1232033       0          0  1
2     0    6     1   3.835729 0.2546201       0          0  2
3     0    1     0   6.297057 0.2655715       0          0  3
4     1   16     1   6.297057 0.2655715       0          1  3
5     0   36     0  -7.737166 0.4900753       0          0  4
6    36   39     1  -7.737166 0.4900753       0          1  4
```

- Bootstrap estimates are similar to Classical estimates, but provide additional variability insight.
- Bayesian estimation offers more information through posterior distributions, including credible intervals.

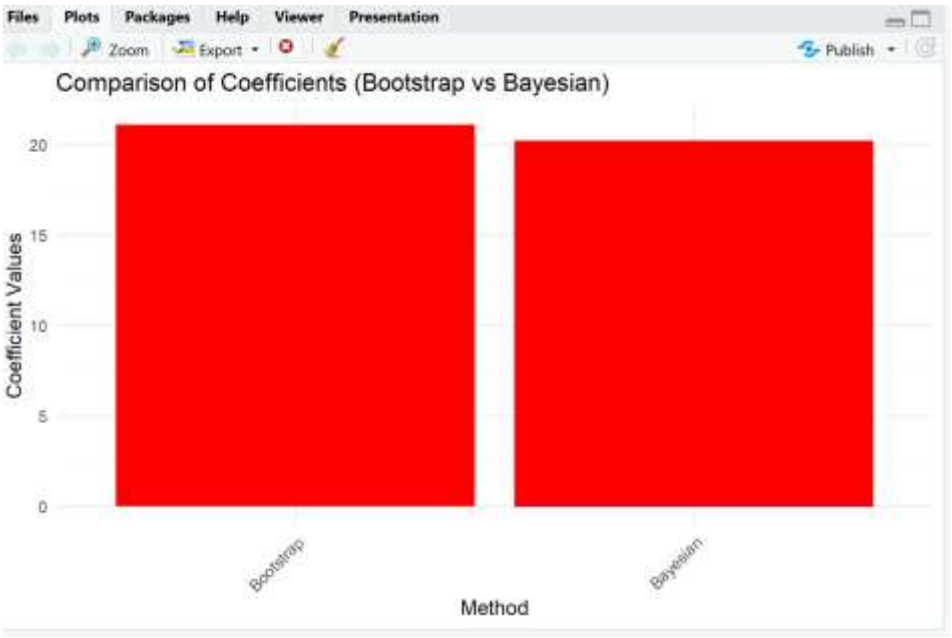

Figure 1: Comparison of Coefficient Estimates

5. Conclusion

In the matched case–control framework that supports the conditional logistic regression model, the analyses illustrated in the attached study provide evidence that Bootstrap and Bayesian methods are in general valid and produce coefficient estimates on key predictors like age and transplant status that are very comparable. The results show that the Bootstrap method gives fairly reliable point estimates and has a practical advantage in providing resampling based measures of variability that are appropriate for more applied situations with limited capacity for computing or modeling complexity. Nevertheless, together with the computational challenges noted in the current study, Bayesian estimation shows promise by providing richer inferential output (i.e., posterior distributions and credible intervals) and greater interpretability and flexibility (><'s, K.E.; me–xs, C.A.) in complex modeling cases. The close agreement between the two approaches highlights the robustness of the estimated parameters and supports the continued use of Bootstrap techniques as a reliable alternative to Bayesian inference in routine analyses. Our results, thus, indicate that for short-term analysis or in scenarios where resources are limited, researchers can safely use Bootstrap methods, while Bayesian frameworks are still beneficial for more profound and extensive analysis that relies on prior data and seek deeper probabilistic understanding. Future work will be needed to enhance implementation of the Bayesian model, explore sensitivity to prior specifications, and extend the comparison to larger datasets and more complex conditional regression structures to further evaluate the scalability and inferential advantages of both methods.

REFERENCES

[1] T. M. Therneau, *survival: Survival Analysis*, R package version 3.8-6, 2023. [Online]. Available: https://CRAN.R-project.org/package=survival

[2] A. C. Davison and D. V. Hinkley, *Bootstrap Methods and Their Application*. Cambridge, U.K.: Cambridge Univ. Press, 1997.

[3] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, *Bayesian Data Analysis*, 3rd ed. Boca Raton, FL, USA: CRC Press, 2020.

[4] I. T. Jolliffe, *Principal Component Analysis*. New York, NY, USA: Springer, 2002.

[5] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008.

[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[7] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.
[8] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *J. R. Stat. Soc. B*, vol. 61, no. 3, pp. 611–622, 1999.
[9] Z. Zhao and J. Liu, *Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management*. Hoboken, NJ, USA: Wiley, 2015.
[10] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.
[11] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.
[12] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
[13] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," *Adv. Neural Inf. Process. Syst.*, vol. 14, pp. 585–591, 2002.
[14] D. L. Donoho and C. Grimes, "Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data," *Proc. Natl. Acad. Sci. USA*, vol. 100, no. 10, pp. 5591–5596, 2003.
[15] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.