Bayesian Lasso logistic regression

Wei Lu

Basic information

Team member: Wei Lu

Github Link:

Theme: Comparison between Bayesian classification models

Compare the simple Logistic regression and Lasso logistic regression with spike-and-slab priors.

Motivation: The classic Lasso regression performs feature selections inherently by shrinking coefficients to zero, but it is difficult to evaluate the significance of these coefficients. Instead, Bayesian Lasso provides an alternative to measure the uncertainty of parameters by credible intervals [1]. There are many types of priors on parameters such as the Laplace priors, the adaptive priors, the spike-and-slab priors, etc [2]. This project will focus on spike-and-slab priors and use the logistic link function to perform classification on the Pumpkin Seeds Dataset. Previous study shows that the frequentist's logistic regression achieves 87.92% accuracy on the same dataset[3]. This project aims to develop a calibrated hierarchical Lasso model to accurately classify the pumpkin seeds with less features. In addition, the posterior distributions of selected/unselected parameters under different situations (e.g. mis-specification, small sample size) will be investigated compared to the baseline model.

Potential approaches:

First, we split the dataset into training data and testing data. Exploratory data analysis will be be performed on the training data, followed by data preprocessing. A simple Bayesian Logistic model will work as an baseline for comparison. The Lasso logistic model will be built following the recipe and references. After the initial evaluation, potential studies might be: (1) to further decrease the prior sensitivity. (2) to investigate the performance of models and the selected features when the data size becomes smaller.

Candidate datasets:

Main dataset: https://www.kaggle.com/datasets/muratkokludataset/pumpkin-seeds-dataset/data

```r
suppressPackageStartupMessages(require(readxl))
pumpkin_df <- read_xlsx("Data/Pumpkin_Seeds_Dataset.xlsx")
head(pumpkin_df)
```

```
# A tibble: 6 x 13
   Area Perimeter Major_Axis_Length Minor_Axis_Length Convex_Area Equiv_Diameter
  <dbl>     <dbl>             <dbl>             <dbl>       <dbl>          <dbl>
1 56276      888.              326.              220.       56831           268.
2 76631     1068.              417.              234.       77280           312.
3 71623     1083.              436.              211.       72663           302.
4 66458      992.              382.              223.       67118           291.
5 66107      998.              384.              220.       67117           290.
6 73191     1041.              406.              231.       73969           305.
# i 7 more variables: Eccentricity <dbl>, Solidity <dbl>, Extent <dbl>,
#   Roundness <dbl>, Aspect_Ration <dbl>, Compactness <dbl>, Class <chr>
```

Backup dataset: https://www.kaggle.com/datasets/erdemtaha/cancer-data/data

```r
cancer_df <- read.csv("Data/Cancer_Data.csv")
head(cancer_df)
```

```
        id diagnosis radius_mean texture_mean perimeter_mean area_mean
1   842302         M       17.99        10.38         122.80    1001.0
2   842517         M       20.57        17.77         132.90    1326.0
3 84300903         M       19.69        21.25         130.00    1203.0
4 84348301         M       11.42        20.38          77.58     386.1
5 84358402         M       20.29        14.34         135.10    1297.0
6   843786         M       12.45        15.70          82.57     477.1
  smoothness_mean compactness_mean concavity_mean concave.points_mean
1         0.11840          0.27760         0.3001             0.14710
2         0.08474          0.07864         0.0869             0.07017
3         0.10960          0.15990         0.1974             0.12790
4         0.14250          0.28390         0.2414             0.10520
5         0.10030          0.13280         0.1980             0.10430
6         0.12780          0.17000         0.1578             0.08089
  symmetry_mean fractal_dimension_mean radius_se texture_se perimeter_se
```

```
1        0.2419                0.07871    1.0950     0.9053         8.589
2        0.1812                0.05667    0.5435     0.7339         3.398
3        0.2069                0.05999    0.7456     0.7869         4.585
4        0.2597                0.09744    0.4956     1.1560         3.445
5        0.1809                0.05883    0.7572     0.7813         5.438
6        0.2087                0.07613    0.3345     0.8902         2.217
  area_se smoothness_se compactness_se concavity_se concave.points_se
1  153.40      0.006399        0.04904      0.05373           0.01587
2   74.08      0.005225        0.01308      0.01860           0.01340
3   94.03      0.006150        0.04006      0.03832           0.02058
4   27.23      0.009110        0.07458      0.05661           0.01867
5   94.44      0.011490        0.02461      0.05688           0.01885
6   27.19      0.007510        0.03345      0.03672           0.01137
  symmetry_se fractal_dimension_se radius_worst texture_worst perimeter_worst
1     0.03003             0.006193        25.38         17.33          184.60
2     0.01389             0.003532        24.99         23.41          158.80
3     0.02250             0.004571        23.57         25.53          152.50
4     0.05963             0.009208        14.91         26.50           98.87
5     0.01756             0.005115        22.54         16.67          152.20
6     0.02165             0.005082        15.47         23.75          103.40
  area_worst smoothness_worst compactness_worst concavity_worst
1     2019.0           0.1622            0.6656          0.7119
2     1956.0           0.1238            0.1866          0.2416
3     1709.0           0.1444            0.4245          0.4504
4      567.7           0.2098            0.8663          0.6869
5     1575.0           0.1374            0.2050          0.4000
6      741.6           0.1791            0.5249          0.5355
  concave.points_worst symmetry_worst fractal_dimension_worst  X
1               0.2654         0.4601                 0.11890 NA
2               0.1860         0.2750                 0.08902 NA
3               0.2430         0.3613                 0.08758 NA
4               0.2575         0.6638                 0.17300 NA
5               0.1625         0.2364                 0.07678 NA
6               0.1741         0.3985                 0.12440 NA
```

References:

[1]: Park, T., & Casella, G. (2008). The Bayesian Lasso. Journal of the American Statistical Association, 103(482), 681–686. https://doi.org/10.1198/016214508000000337

[2]: Chen, S. M., Bauer, D. J., Belzak, W. M., & Brandt, H. (2021). Advantages of Spike and Slab Priors for Detecting Differential Item Functioning Relative to Other Bayesian Regularizing

Priors and Frequentist Lasso. *Structural Equation Modeling: A Multidisciplinary Journal*, *29*(1), 122–139. https://doi.org/10.1080/10705511.2021.1948335

[3]: KOKLU, M., SARIGIL, S., & OZBEK, O. (2021). The use of machine learning methods in classification of pumpkin seeds (Cucurbita pepo L.). Genetic Resources and Crop Evolution, 68(7), 2713-2726. Doi: https://doi.org/10.1007/s10722-021-01226-0