P8160 Group Project 2: Breast Cancer Diagnosis and Optimizations

Wenhan Bao | Tianchuan Gao | Jialiang Hua | Qihang Wu | Paula Wu

3/25/2022

Objective

Background & Methods

Background

Methods

Exploratory Data Analysis (EDA)

Modified Newton-Raphson Algorithm

Logistic-LASSO

Least Absolute Shrinkage and Selection Operator (LASSO) To estimate coefficients through Newton-Raphson method, it is necessary to compute the corresponding inverse of Hessian Matrix $[\nabla^2 f(\theta_{i-1})]^{-1}$. However, the computational burden of calculation will increase as the dimension of predictors increases and the collinearity will also be a problem. Therefore, we use a regularization method, LASSO, to shrink coefficients and perform variable selections. For regression lasso, the objective function is:

$$f(\beta) = \frac{1}{2}\sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} x_{i,j}\beta_j)^2 + \lambda \sum_{i=1}^{p} |\beta_j|,$$

where the first term is residual sum of squares (RSS) and the second term is the lasso l1 penalty. Noted that the $x_{i,j}$ needs to be standardized before LASSO so that the penalty will be equally applied to all predictors. For each single predictor, the LASSO solution is like:

$$\hat{\beta}^{lasso}(\lambda) = S(\hat{\beta}, \lambda) = \begin{cases} \hat{\beta} - \lambda, \text{ if } \hat{\beta} > 0 \text{ and } \lambda < |\hat{\beta}| \\ \hat{\beta} + \lambda, \text{ if } \hat{\beta} < 0 \text{ and } \lambda < |\hat{\beta}| \\ 0, \text{ if } \lambda > |\hat{\beta}|, \end{cases}$$

where $S(\hat{\beta}, \lambda)$ is called soft threshold. The basic idea of this function is to shrink all β coefficients between $-\lambda$ and λ.

Coordinate-wise Descent Algorithm Another approach to solve the complex computation of inverse Hessian Matrix is coordinate descent approach, which starts with an initial guess of parameters θ, optimize one parameter at one time based on the best knowledge of other parameters, and use the results as the start values for the next iteration. We then repeat the above steps until convergence. Finally, this approach tries to minimize the following objective function when considering a lasso penalty:

$$f(\beta_j) = \frac{1}{2}\sum_{i=1}^{n}(y_i - \sum_{k \neq j} x_{i,j}\tilde{\beta}_k - x_{i,j}\beta_j)^2 + \lambda \sum_{k \neq j} |\tilde{\beta}_k| + \lambda|\beta_j|,$$

where $\tilde{\beta}$ represents the current estimates of β and therefore constants. If we also consider a weight ω_i is associated with each observation, then the updated β_j in this case is:

$$\tilde{\beta}_j(\lambda) \leftarrow \frac{S(\sum_i \omega_i x_{i,j}(y_i - \tilde{y}_i^{(-j)}), \lambda)}{\sum_i \omega_i x_{i,j}^2},$$

where $\tilde{y}_i^{(-j)} = \sum_{k \neq j} x_i \tilde{\beta}_k$. From here, we use the Taylor expansion. So the log-likelihood around "current estimate" is:

$$l(\beta) = -\frac{1}{2n} \sum_{i=1}^{n} \omega_i (z_i - X_i \beta)^2,$$

where working weights $\omega_i = \tilde{\pi}_i(1 - \tilde{\pi}_i)$, working response $z_i = X_i \tilde{\beta} + \frac{y_i - \tilde{\pi}_i}{\tilde{\pi}_i(1-\tilde{\pi}_i)}$, and $\tilde{\pi}_i = \frac{exp(X_i \tilde{\beta})}{1+exp(X_i \tilde{\beta})}$.
Finally, similar to regression lasso, the logistic lasso can be written as a penalized weighted least-squares problem like this:

$$min_\beta L(\beta) = -l(\beta) + \lambda \sum_{j=0}^{p} |\beta_j|$$

.

Pathwise Coordinate-wise Algorithm The difference between pathwise coordinate-wise method and coordinate-wise method is that a sequence value of lambda is required to input. There are some steps taken as followed:

- Select a λ_{max} for all the estimate $\beta = 0$ which is the inner product ($max_l \langle X_l, y \rangle$).
- Compute the solution with a sequence of descending λ from maximum to zero ($\lambda_{max} > \lambda_k > \lambda_{k-1} ... > 0$).
- Initialize coordinate descent algorithms for λ_k by the calculated estimate β from previous λ_{k+1} as a warm start
- By repeating the two steps above, a sequence of optimal coefficients β for each descending λ. When objective function is taken the minimum value ($min_\beta L(\beta)$), the best λ could be identified and optimal coefficients β could be selected under this best λ

5-fold Cross-validation

In a nutshell, a 5-fold cross-validation first splits the shuffled training data (80% of the whole dataset) into 5 parts and takes one group as the hold-out set (validation set) while fitting the model using the remaining 4 training sets. After a model is fitted, we evaluate and retain the model performance, namely AUC and RSS, and move on to the next iteration. After 5 iterations, we calculated the mean RSS and the mean AUC with standard deviations. Our goal is to find an optimal λ that maximizes the mean AUC.

In our implementation, we initially choose 30 equally spaced λ from $e^{(-4,3)}$. The reason why the λ's are widespread is that we would like to determine the rough range, instead of the precise value, of the optimum of this tuning parameter first. For each λ, we ran the 5-fold cross-validation. Within each fold, we use Coordinate-wise optimization to update the coefficients vectors $\boldsymbol{\beta}$ and evaluate the performance by calculating AUC and RSS. Among 30 λ, the optimal λ that maximizes the AUC equals 2.3681. We then narrow down the range to (3, 0) and choose 150 equally spaced λ in between. The best λ selected equals 1.5705.

Comparsion: "Optimal" model vs. "Full" model

The optimal λ selected in the previous section will be compared to the full model, where $\lambda = 0$.

Conclusions

Logistic Regression Model

Logistic-LASSO Model

Discussion

Contributions

We contributed to this project evenly.

Appendix

Reference