

Draft Report

Ngoc Duong, Cui Sitong (sc4636), Xinru Wang(xw2676), Jin Ge

4/15/2020

Objective

Breast cancer is one of the most common cancers in women. However, early diagnoses of breast cancer can aid in reducing the mortality rate. Additionally, advances in imaging technologies and statistical methodologies have allowed for higher-quality data and novel models that could improve the precision of breast cancer diagnoses. The purpose of our project is to build and compare different models in classifying breast cancer tumor as benign or malignant based on image-based predictors. Specifically, we are looking to build a logistic regression model using Newton-Raphson method, and a logistic-LASSO model using coordinate-wise optimization algorithm.

Dataset

There were 569 images collected independently from patients, 212 of whom had malignant tumor and 357 were benign cases. The images were broken down into 30 predictors, corresponding to the mean, standard deviation, and largest values (points on the tails) of the following 10 features:

- radius (mean of distances from center to points on the perimeter)
- texture (standard deviation of gray-scale values)
- perimeter
- area
- smoothness (local variation in radius lengths)
- compactness ($\text{perimeter}^2 / \text{area} - 1.0$)
- concavity (severity of concave portions of the contour)
- concave points (number of concave portions of the contour)
- symmetry
- fractal dimension ("coastline approximation" - 1)

Data cleaning

As shown in the pairwise correlation plot (**Fig. 1**), we can observe the presence of some strong multicollinearity among the predictors. For instance, the `radius_mean` variable has almost perfect correlation of 1 and 0.99 with `perimeter_mean` and `area_mean` variables, respectively. We then left out variables that are correlated by more than 85% with other predictors. The final dataset contained 13 predictors.

Next, considering the LASSO is not scale-invariant, we standardized the design matrix. This is to ensure comparability of estimates by the logistic-LASSO model and Newton-Raphson/logistic regression model. The standardization formula is as follows:

$standardized(x_{ij}) = \frac{x_{ij} - \bar{x}_j}{std(x_j)}$ for $i = 1, 2, \dots, 30$ and $j = 1, 2, \dots, 569$

Finally, we recoded response variable such that “malignancy” = 1, and “benign” = 0.

Newton-Raphson model

We used logistic regression to classify the malignancy of tissue. Malignancy corresponds to response variable being 1 ($y^{(i)} = 1$).

Log likelihood is

$$l(y; \beta) = \sum_{i=1}^n \{y_{(i)} \log \mu_{(i)} + (1 - y_{(i)}) \log(1 - \mu_{(i)})\}$$

Its gradient is given by

$$g : \nabla l(y; \beta) = \sum_{i=1}^n (y_{(i)} - \mu_{(i)}) x_{(i)} = X^T (y - \mu)$$

Its Hessian matrix is given by

$$H : \nabla^2 l(y; \beta) = - \sum_{i=1}^n \mu_{(i)} (1 - \mu_{(i)}) x_{(i)} (x_{(i)})^T = -X^T S X$$

where $S = \text{diag}(\mu_{(i)}(1 - \mu_{(i)}))$, and

$$\mu_{(i)} = p_{\theta}(y = 1|x) = \frac{e^{X_i \beta}}{1 + e^{X_i \beta}}$$

Since we have several predictors, we want to optimize several likelihood functions simultaneously. This is equivalent to solving a system of log-likelihood equations $\nabla l(y; \beta_j) = 0$ where $j = 1, 2, \dots, 13$. To achieve this, we used the Newton Raphson algorithm.

Newton-Raphson Algorithm

Starting at a current point β_i , we can expand the log-likelihood function around this point using Taylor's expansion, which gives a neighborhood of β_i containing β_{i+1} which increases the likelihood. The equation below can be used to iteratively update β_i until the sequence converges and $\nabla l(y; \beta_j) = 0$ is satisfied:

$$\beta_{i+1} = \beta_i - [\nabla^2 l(\beta_i)]^{-1} \nabla l(\beta_i).$$

- Modifications to Newton-Raphson

When implementing Newton-Raphson, we need to check at every step, that the updating direction (for β_{i+1}) is heading to a maximum, and that the point is moving sufficient distances towards the maximum so we do not miss it. Therefore, we also implemented some modifications, specifically gradient descent and step-halving.

- For step-halving, we modified the updating function for β_{i+1} as follows:

$\beta_{i+1} = \beta_i - \lambda [\nabla^2 l(y; \beta_i)]^{-1} \nabla l(y; \beta_i)$, where $\lambda = 1$ until $l(\beta_{i+1}) \leq l(\beta_i)$, which means the new point would have gone too far. Then, we can search for a value λ such that $l(\beta_{i+1}, \lambda) \geq l(\beta_i)$. At this step, we can cut the step, or λ in half for each sub-iteration.

- For gradient descent, at every iteration, we checked whether $\nabla^2 l(y; \beta)$ is negative definite (signifying the point is moving in the right direction). If $\nabla^2 l(y; \beta)$ is not, we replace it with a similar negative definite matrix, such as $\nabla^2 l(y; \beta) - \gamma I$ where γ is chosen such that the resulting matrix is negative definite. Naturally, this γ must be greater than any of the elements of the diagonal matrix D obtained by eigendecomposing $\nabla^2 l(y; \beta) = P^T D P$.

Logistic-LASSO model

The LASSO method aims to minimize the following equation with a penalty term, a quadratic approximation to the negative log likelihood by Taylor expansion around the current estimate, which is:

$$f(\beta) = \frac{1}{2n} \sum_{i=1}^n w_i (z_i - \sum_{j=1}^p x_{i,j} \beta_j)^2 + \lambda \sum_{j=1}^p |\beta_j|, \lambda \geq 0$$

where

- w_i are the working weights, defined as $\tilde{p}(x_i)(1 - \tilde{p}(x_i))$, and p_i is the probability of event for each observation
- z_i are the working response $= \tilde{\beta}_0 + x_i^T \tilde{\beta} + \frac{y_i - \tilde{p}(x_i)}{\tilde{p}(x_i)(1 - \tilde{p}(x_i))}$, working response

Then, each β_i is optimized using the following equation:

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

where $S(\hat{\beta}, \gamma)$ is called soft-threshold and is defined as $S(\hat{\beta}, \lambda) = \text{sign}(\hat{\beta})(|\hat{\beta}| - \lambda)_+$

Coordinate-wise Descent algorithm is used to minimize this function. We can form a quadratic approximation to the negative log likelihood by Taylor expansion around the current estimate, which is:

$$f(\beta) = -\frac{1}{2n} \sum_{i=1}^n w_i (z_i - \sum_{j=0}^p x_{i,j} \beta_j)^2 + C(\tilde{\beta})$$

where

$$z_i = \tilde{\beta}_0 + x_i^T \tilde{\beta} + \frac{y_i - \tilde{p}(x_i)}{\tilde{p}(x_i)(1 - \tilde{p}(x_i))}, \text{ working response}$$

$$w_i = \tilde{p}(x_i)(1 - \tilde{p}(x_i)), \text{ working weights}$$

and $\tilde{p}(x_i)$ is evaluated at the current parameters, the last term is constant. The Newton update is obtained by minimizing the $f(\beta)$

The coordinate descent algorithm to solve the penalized weighted least squares problem

$$\max_{\beta \in \mathbb{R}^{p+1}} \{f(\beta) + \lambda P(\beta)\}$$

Results

The coefficient estimates can be found in (**Table 1**). Newton-Raphson algorithm gives quite similar estimates to those in the logistic regression model produced by GLM package. For the logistic-LASSO model, we can see the coefficient estimates are approximately close to the ones produced by GLMnet with 5-fold cross-validation. They do not exactly match, however, due to the potential dissimilarities in set-up conditions in our implementation and theirs.

The path of solution could be found in (**Figure 3**), and the distribution of cross-validated MSEs produced by the hand-built logistic-LASSO model could be found in (**Figure 4**). Five-fold cross-validation suggested the best λ is 0.00454, which corresponds to the lowest cross-validated MSE. A similar distribution of cross-validated MSEs produced by the GLMNet package in R can be found in (**Fig. 4**). Here, 5-fold cross-validation suggested the best λ is 0.0037.

Lastly, we wanted to compare the prediction performance using MSE as a criteria. We examined this under two scenarios, one is with 5-fold cross-validation, and the other is 5-fold repeated cross-validation

(number of repeats is 5). These distributions can be found in Figure 3 and 4, respectively. We noticed that the cross-validated MSE of both Newton Raphson and logistic-LASSO are similiary distributed, although the latter seems to perform slightly less well. On the other hand, GLMnet gives the most consistently well performance. Nonetheless, the errors were all in close proximity with one another so we are confident they all very good discriminatory power.

Conclusions

The report aimed to explore how different models perform at the same task of classifying breast cancer tumors into benign and malignant types using various predictors derived from the tumor images. Since we have eliminated most multicollinearity at the beginning, it is reasonable to expect Newton-Raphson and logistic-LASSO to have quite similar discriminatory performance. On the other hand, we would expect to see logistic-LASSO to perform better than Newton Raphson in terms of predictive ability in the presence of higher-dimensional data, and with more correlated predictors. We will aim to explore this further as more time and bandwidth allow. All in all, these models we looked at in this report nevertheless perform decently in classifying correctly each type of breast cancer.

Appendix

Figure

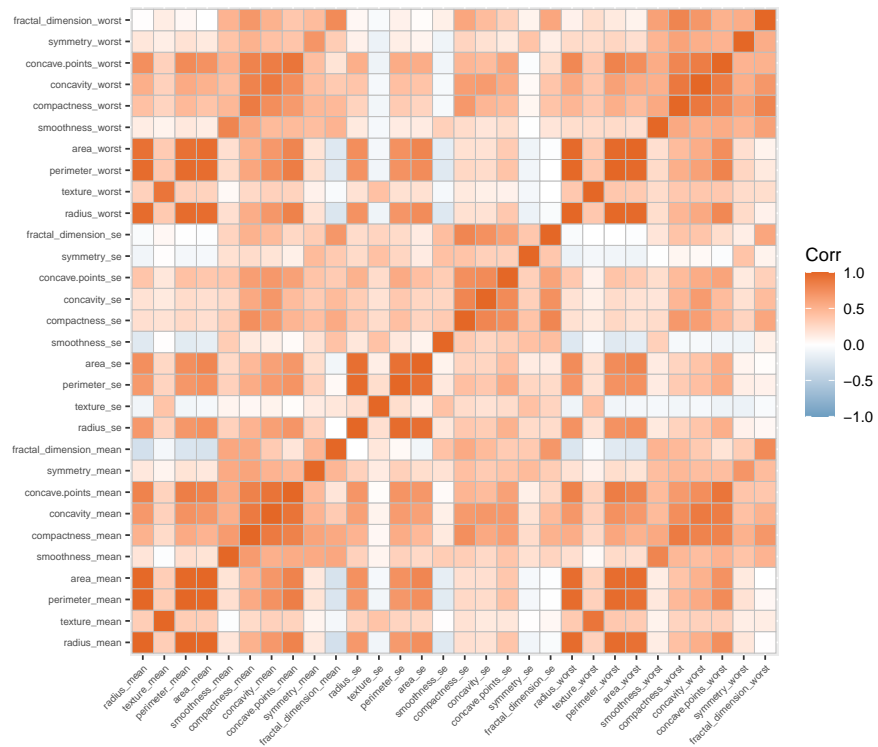


Figure 1: Multicollinearity plot of the dataset

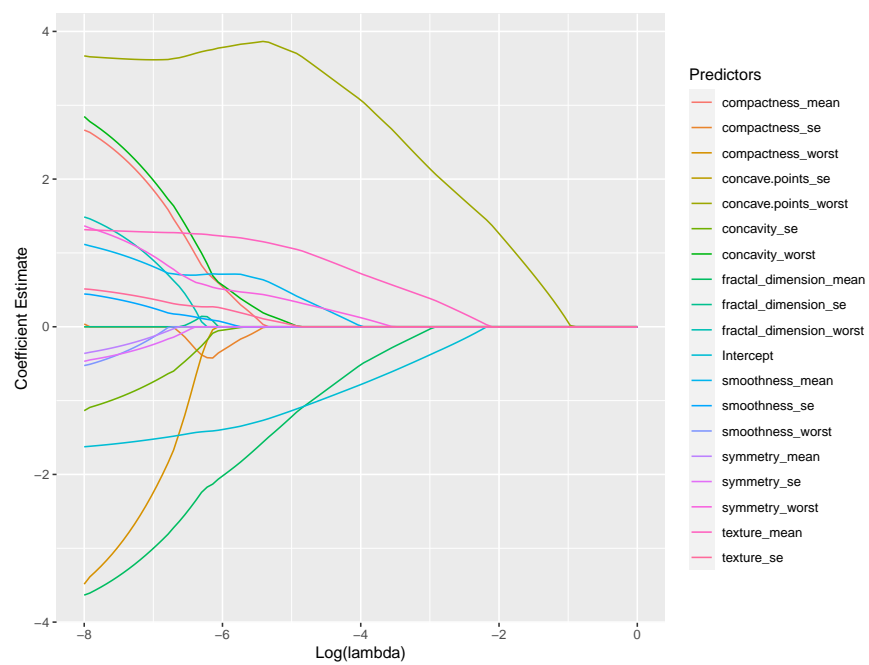


Figure 2: The path of solution

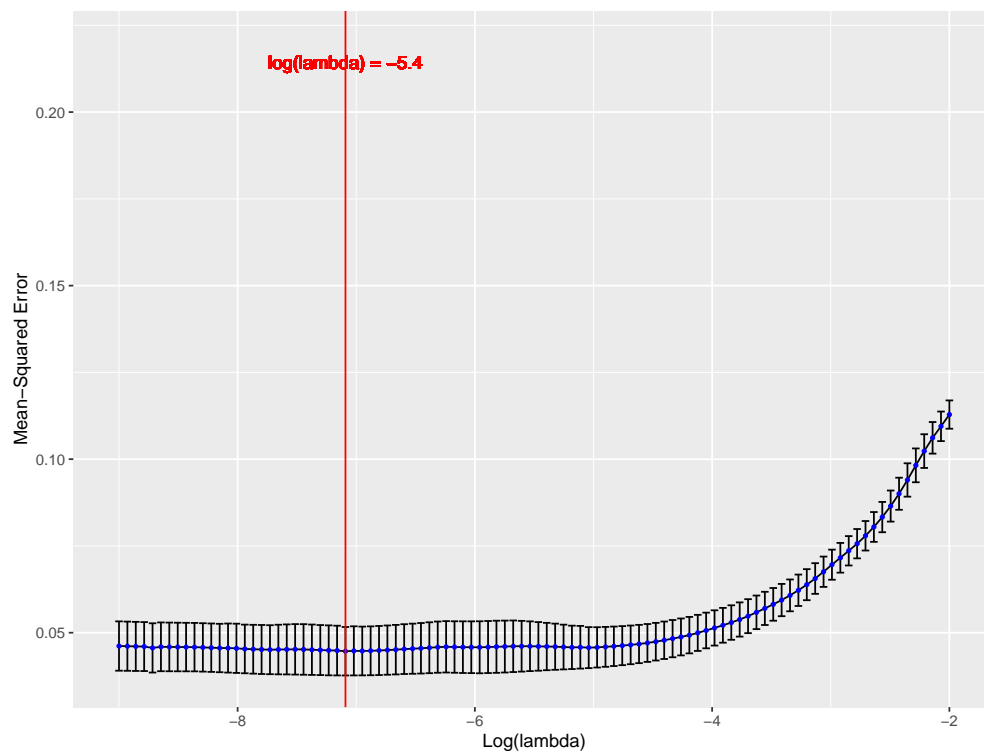


Figure 3: Plot of cross-validated MSEs by hand-built model

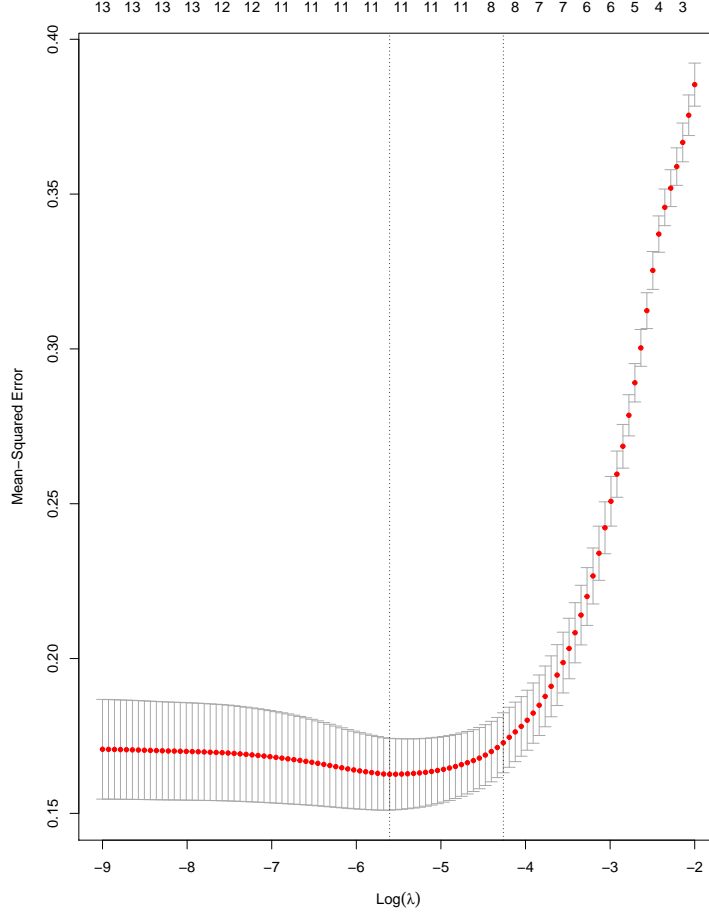


Figure 4: Plot of cross-validated MSEs by the GLMNet

	GLM binomial	Newton-Raphson	GLMnet	Logistic-LASSO
smoothness_mean	1.5725	1.5725	1.1176	1.0229
symmetry_mean	-0.1187	-0.1187	0.0000	0.0000
fractal_dimension_mean	-4.5061	-4.5061	-3.5738	-3.3703
texture_se	0.8068	0.8068	0.5009	0.4466
smoothness_se	-0.8304	-0.8304	-0.6757	-0.6253
compactness_se	0.3954	0.3954	0.1787	0.1392
concavity_se	-0.0300	-0.0300	0.0000	0.0000
concave.points_se	2.2856	2.2856	1.8469	1.7472
symmetry_se	-0.2039	-0.2039	-0.0997	-0.0615
fractal_dimension_se	-0.7551	-0.7551	-0.3363	-0.2821
smoothness_worst	0.7859	0.7859	0.8685	0.8577
symmetry_worst	1.0091	1.0091	0.7511	0.7050
fractal_dimension_worst	2.8346	2.8346	2.1066	1.9924
intercept	-1.5278	-1.5278	-1.2742	-1.1624

Figure 5: The coefficient estimates of each model