

# Project 2: Breast Cancer Diagnosis

Shengzhi Luo, Xinran Sun, Haotian Wu, Lin Yang

3/31/2022

## Objectives

A mammogram is an X-ray image of breast tissue. It can help save lives because it is easier to treat breast cancer in its early stages before the cancer is big enough to detect or cause symptoms. However, a wrong diagnosis can have a negative impact on patients. For example, if there is a false-positive test result, the doctor sees something that looks like cancer but is not. This could result in overtreatment that causes unnecessary side effects on patients. On the other hand, false-negative test result occurs when a doctor misses cancer tissues, which may delay the treatment. Therefore, building a model that gives an accurate classification of the tissue images is necessary to give proper treatment. In our study, we collected 569 images from both malignant and benign cancer tissues. Our goal is to build a predictive model to facilitate cancer diagnosis.

## Dataset

Our data set consists of 569 rows, with 357 benign and 212 malignant. We denote 0 for benign and 1 for malignant. We also have 30 columns representing the features of the tissue images. They include the mean, standard deviation, and the largest values of the distributions of the following 10 features computed for the cell nuclei:

- 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 ( $perimeter^2/area - 1.0$ )
- concavity (severity of concave portions of the contour)
- concave points (number of concave portions of the contour)
- symmetry
- fractal dimension ("coastline approximation" - 1)

## EDA

Before building the model, we want have a close look at the dataset. Therefore, we first examine the correlation between variables. The squares with dark color in the correlation plot has strong correlation with each other. We can see there is very strong correlation between radius, perimeter and area across mean, standard deviation, and the largest values. We decided to drop some variables with correlation larger than 0.7. The variables we dropped are perimeter\_mean, area\_mean, compactness\_mean, concave\_points\_mean, perimeter\_se, area\_se, radius\_worst, texture\_worst, perimeter\_worst, area\_worst, and concavity\_worst (11 variables).

After that, we built feature plot to analyze the relationship between variables after removing the variables with high correlation. From this plot, we can see that there are no strong relationship between variables after removing. We also found that the points for benign tissues are often locate at left-bottom side, which indicates the benign tissues usually have smaller feature values compared to malignant tissues.

We also calculated the mean of each variables to compare values between benign and malignant cases. According to the average values of the mean of each feature, we can find that benign tissues have smaller values compared to malignant tissues, except for fractal dimension. There is no general pattern for the average values of the standard deviations. Based the average values of the largest value of each feature, we can find that benign tissues have smaller largest values compared to malignant tissues.

To compare prediction performance of different models, the dataset is partitioned into the training data (0.8) and the test data (0.2).

## Methods

### Logistic Regression Model

Let  $y$  be the vector with 569 binary response variable,  $X$  be the  $569 \times 30$  matrix with 30 numerical explanatory variables, and  $\beta$  be the vector with 30 corresponding coefficients. We also have  $\beta_0$  as the intercepts.

For our logistic model, the probability of  $i$ th row be a malignant tissue is given by:

$$P(y_i = 1|X_i) = \frac{e^{\beta_0 + \beta X_i}}{1 + e^{\beta_0 + \beta X_i}}.$$

For likelihood function is:

$$L(\beta_0, \beta) = \prod_{i=1}^n \left[ \left( \frac{e^{\beta_0 + \beta X_i}}{1 + e^{\beta_0 + \beta X_i}} \right)^{y_i} \left( \frac{1}{1 + e^{\beta_0 + \beta X_i}} \right)^{1-y_i} \right].$$

Maximizing the likelihood is equivalent to maximizing the log likelihood:

$$f(\beta_0, \beta) = \sum_{i=1}^n [y_i(\beta_0 + \beta X_i) - \log(1 + e^{\beta_0 + \beta X_i})].$$

The gradient of this function is:

$$\nabla f(\beta_0, \beta) = \begin{pmatrix} \sum_{i=1}^n y_i - p_i \\ \sum_{i=1}^n X_1(y_i - p_i) \\ \dots \\ \sum_{i=1}^n X_n(y_i - p_i) \end{pmatrix} = X^T(y_i - p_i)$$

where  $p_i = P(y_i = 1|X_i)$  as mentioned in previous probability function.

The Hessian is given by

$$\nabla^2 f(\beta_0, \beta) = -X^T W X$$

where  $W = p_i(1 - p_i)$ .

## Newton-Raphson Algorithm

### Path-wise Coordinate-wise Optimization Algorithm for Logistic-Lasso Model

To obtain a path of coefficients for a descending sequence of tuning parameter  $\lambda$ , we need to develop a coordinate-wise descent algorithm estimating coefficients for a specific lambda. The logistic-lasso can be written as a penalized weighted least-squares problem:

$$\min_{(\beta_0, \beta_1)} L(\beta_0, \beta_1, \lambda) = \left\{ -\ell(\beta_0, \beta_1) + \lambda \sum_{j=0}^p |\beta_j| \right\}$$

When there are a large number of parameters, i.e.,  $p$  is large, a coordinate-wise descent algorithm is required to optimize coefficients. The objective function is:

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

Minimizing  $f(\beta_j)$  w.r.t  $\beta_j$  while having  $\tilde{\beta}_k$  fixed, we have weighted updates to update one coefficient at a time iteratively until the log-likelihood converges:

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

where  $\tilde{y}_i^{(-j)} = \sum_{k \neq j} x_{i,k} \tilde{\beta}_k$ .

If we apply Taylor expansion to the log-likelihood around “current estimates”  $(\tilde{\beta}_0, \tilde{\beta}_1)$ , we have a quadratic approximation function  $f(\beta_0, \beta_1)$  to the log-likelihood:

$$f(\beta_0, \beta_1) \approx \ell(\beta_0, \beta_1) = -\frac{1}{2n} \sum_{i=1}^n w_i (z_i - \beta_0 - \mathbf{x}_i^T \beta_1)^2 + C(\tilde{\beta}_0, \tilde{\beta}_1)$$

where

$$z_i = \tilde{\beta}_0 + \mathbf{x}_i^T \tilde{\beta}_1 + \frac{y_i - \tilde{p}_i(\mathbf{x}_i)}{\tilde{p}_i(\mathbf{x}_i)(1 - \tilde{p}_i(\mathbf{x}_i))}$$

$$w_i = \tilde{p}_i(\mathbf{x}_i)(1 - \tilde{p}_i(\mathbf{x}_i))$$

$$\tilde{p}_i = \frac{\exp(\tilde{\beta}_0 + \mathbf{x}_i^T \tilde{\beta}_1)}{1 + \exp(\tilde{\beta}_0 + \mathbf{x}_i^T \tilde{\beta}_1)}$$

$w_i$  is the working weight,  $z_i$  is the working response,  $p_i$  is the probability of malignant case estimated at current coefficients. This quadratic approximation function is used in the coordinate-wise descent algorithm.

We then can develop a path-wise coordinate-wise optimization algorithm to get a path of solutions for a descending sequence of  $\lambda$ .

- Step 1: Find the smallest value  $\lambda$  for which all the estimated  $\beta$  are 0, defined as  $\lambda_{max}$ .

- Step 2: Define a fine sequence  $\lambda_{max} \geq \lambda_1 \geq \dots \lambda_{min} \geq 0$ .
- Step 3: To estimate coefficients of the current  $\lambda_{k+1}$ , implement coordinate descent algorithm using the computed coefficients of the previous  $\lambda_k$  (warm start) as coefficient start values. ( $\lambda_{k+1} < \lambda_k$ )

## Cross Validation

To select the best  $\lambda$  for the optimal model, a 5-fold cross-validation is performed.

- Step 1: We shuffle the original dataset randomly.
- Step 2: Based on the new dataset, we split it into 5 even groups.
- Step 3: From the 5 groups, we take one group as the test set and the remaining groups as training set. And repeat this procedure until all groups are tested.
- Step 4: We fit the model and evaluate on the test set.

## Results

The coefficients of the full logistic regression model based on the training data using `glm()` are shown in **Table 1**. 5 predictors are found to be significant with p values less than 0.05, including **texture\_mean**, **concavity\_mean**, **radius\_se**, **symmetry\_worst**. For the logistic-lasso model, the  $\lambda_{max}$  is calculated to be 175.62, we then defined a descending sequence of 50  $\lambda$  between  $\lambda_{max}$  and  $e^{-5}$ . Since the outcome is a binary variable, AUC is used as the evaluation metric to compare models. Through the 5-fold cross-validation, the best  $\lambda$  is found to be 0.981 with an average AUC of 0.997 (**Figure 3**). The coefficients for the best  $\lambda$  are shown in **Table 2**. As expected, the optimal logistic-lasso model shrinks some coefficients to 0.

AUC scores of the full logistic regression model and the logistic-lasso model are computed based on the test data to compare prediction performance of the two models. The full model's AUC (0.977) is found to be less than the lasso model's AUC (0.996), thus the optimal lasso model slightly outperforms the full model.

## Conclusions

### Findings

The primary goal of our project is to build a model to predict a breast tissue sample is benign or malignant. After the exploratory data analysis and collinearity, we dropped 11 highly correlated variables, and 19 variables are remained for full model. We compared a logistic regression model with all variables estimated from Newton Raphson algorithm against an optimal model estimated by LASSO variable selection and path-wise coordinate-wise optimization algorithm, and found that the more optimal model is LASSO which has less predictors and higher AUC value.

### Limitations

We found that Newton Raphson algorithm is unstable during our work procedure. First, the convergence is not guaranteed. Also, it is dependent on the choice of starting value. If we set initial beta at a large value, the function would take no action. Thus, we have to carefully choose a relatively small beta, for example, 0.001 would be an ideal starting point. Besides lambda selection from cross validation is limited, due to intensive computation of cross validation, we chose 50 lambdas to estimate the best value. However, if we have more lambdas, the result of estimation would be more accurate and close to the truth.