

Breast Cancer Diagnosis using Optimization

Margaret Gacheru, Alyssa Vanderbeek, Jack Yan, Mengyu Zhang

Introduction

In this project, we build a model to classify breast tissue images into malignant cancer or benign tissue. We are interested in observing how the certain features for individual breast tissue are influential in determining cancer diagnosis. Prediction algorithms are potentially useful in this area as a way to aid in the diagnosis of breast cancer.

Our data is taken from tissue imaging from 569 breast cancer patients. Our goal was to build a prediction algorithm from the 30 variables provided. First, we built a full logistic model with all 30 features and utilized a Newton-Raphson algorithm in order to estimate the model coefficients. Furthermore, we built a logistic-lasso model to perform variable selection in order to identify the best subset of tissue characteristics that play a major role in prediction.

Methods

The dataset contains 569 individuals with 30 variables describing the tissue images. These variables correspond to mean, standard deviation, and largest value of 10 tissue characteristics, including, radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, and fractal dimension.

Logistic Regression Model

The logistic regression model can be defined as

$$\log\left(\frac{\pi}{1-\pi}\right) = X\beta$$

where the link function is $\log\left(\frac{\pi}{1-\pi}\right)$.

The likelihood function is

$$L(\beta; X, y) = \prod_{i=1}^n \left\{ \left(\frac{\exp(X_i\beta)}{1 + \exp(X_i\beta)} \right)^{y_i} \left(\frac{1}{1 + \exp(X_i\beta)} \right)^{1-y_i} \right\}$$

where $y_i \sim \text{bin}(1, \pi_i)$,

$$y_i = \begin{cases} 1, & \text{malignant tissue} \\ 0, & \text{benign tissue} \end{cases}$$

Consequently, the log-likelihood function is

$$l(\beta) = \sum_{i=1}^n \{y_i (X_i\beta) - \log(1 + \exp(X_i\beta))\}$$

Maximizing log-likelihood function is equivalent to maximizing likelihood function, so the gradient and Hessian matrix will be

$$\nabla l(\beta) = \begin{pmatrix} \sum_{i=1}^n y_i - p_i \\ \sum_{i=1}^n \mathbf{x}_i (y_i - p_i) \end{pmatrix}_{(p+1) \times 1}$$

and

$$\begin{aligned} \nabla^2 l(\beta) &= - \sum_{i=1}^n \begin{pmatrix} 1 \\ \mathbf{x}_i \end{pmatrix} \begin{pmatrix} 1 & \mathbf{x}_i^T \end{pmatrix} p_i (1 - p_i) \\ &= - \begin{pmatrix} \sum p_i (1 - p_i) & \sum \mathbf{x}_i^T p_i (1 - p_i) \\ \sum \mathbf{x}_i p_i (1 - p_i) & \sum \mathbf{x}_i \mathbf{x}_i^T p_i (1 - p_i) \end{pmatrix} \end{aligned}$$

where $p_i = P(Y_i = 1 | \mathbf{x}_i) = \frac{\exp(\beta_0 + \mathbf{x}_i^T \beta_1)}{1 + \exp(\beta_0 + \mathbf{x}_i^T \beta_1)}$.

With $p = 30$ predictors, we obtain a 31×1 gradient vector and 31×31 Hessian matrix

$$\nabla l(\beta) = \mathbf{X}^T (\mathbf{Y} - \boldsymbol{\pi}), \quad \boldsymbol{\pi} = \frac{e^{\mathbf{X}\beta}}{1 + e^{\mathbf{X}\beta}}$$

where \mathbf{X} is the design matrix

$$H = \mathbf{X}^T \boldsymbol{\Lambda} \mathbf{X}$$

where $\boldsymbol{\Lambda}$ is a diagonal matrix with $\text{var}(y_i)$, $i = 1, \dots, n$ as the diagonal elements

Algorithms

Newton-Raphson Algorithm

Newton-Raphson algorithm is a method to search for solutions to the system of equations $\nabla l(\beta) = 0$. At each step, given the current point β_0 , the gradient $\nabla l(\beta)$ for β near β_0 may be approximated by

$$\nabla l(\beta_0) + \nabla^2 l(\beta_0) (\beta - \beta_0)$$

The next step in the algorithm is determined by solving the system of linear equations

$$\nabla l(\beta_0) + \nabla^2 l(\beta_0) (\beta - \beta_0) = \mathbf{0}$$

and the next “current point” is set to be the solution, which is

$$\beta_1 = \beta_0 - [\nabla^2 l(\beta_0)]^{-1} \nabla l(\beta_0)$$

The i th step is given by

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

Coordinate-wise descent with regularized logistic regression

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 the p , the number of parameters, is large, the optimization could be challenging. Therefore, a coordinate-wise descent algorithm will be applied. 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 β_j while having $\tilde{\beta}_k$ fixed, we have weighted beta updates

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

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

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

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

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

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

Therefore, the algorithm's steps are

Step 1. Find λ_{max} such that all the estimated β are zero;

Step 2. Define a fine sequence $\lambda_{max} \geq \lambda_1 \geq \dots \geq \lambda_{min} \geq 0$;

Step 3. Defined the quadratic approximated objective function $L(\beta_0, \beta_1, \lambda)$ for λ_k using the estimated parameter at λ_{k-1} ($\lambda_{k-1} > \lambda_k$).

Step 4. Run a coordinate-wise descent algorithm to optimize the function defined in Step 3.

Results