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 mortatlity 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 benignant or malignant based on image-based predictors. Specifically, we are look 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 (perimeter2 / 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 multi-collinearity 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 $i = 1, 2, ...30$ and $j = 1, 2, ..., 569$

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

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

Its Hessian matrix is given by

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

where $S = diag(\mu_{(i)}(1 - \mu_{(i)}))$, 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 likehood functions simultaneously. This is equivalent to solving a system of log-likelihood equations $\bigtriangledown l(y; \beta_j) = 0$ where $j = 1, 2, ...13$. 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 $\bigtriangledown l(y; \beta_j) = 0$ is satisfied:

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

- 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:

$\beta_{i+1} = \beta_i - \lambda[\bigtriangledown^2 l(y; \beta_i)]^{-1} \bigtriangledown l(y; \beta_i)$, 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 $\bigtriangledown^2 l(y; \beta)$ is negative definite (signifying the point is moving in the right direction). If $\bigtriangledown^2 l(y; \beta)$ is not, we replace it with a similar negative definite matrix, such as $\bigtriangledown^2 l(y; \beta) - \gamma I$ 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 $\bigtriangledown^2 l(y; \beta) = P^T D P$.