STATS 212: Generalized Linear Models
Lecture 4: Computational Methods; Binary Response Variable

Babak Shahbaba

UCI, Spring 2010

Numerical methods for finding MLE

- So far, we have discussed generalized linear models in the context of exponential families.
- We saw that the likelihood equation for these models takes a simple form, especially if we use canonical links.
- However, the likelihood equations are in general nonlinear in β, and as the result, numerical methods are needed to find $b\hat{e}ta$.
- In what follows, we will discuss some of these methods.

Newton-Raphson method

- Newton-Raphson method is a general purpose iterative algorithm for solving nonlinear equations.
- We would use this method to solve likelihood equations.
- Denote the log-likelihood as $L(\beta)$. Our objective is to find the value of β for which $L(\beta)$ is maximized.
- We start with the single parameter case.

Newton-Raphson method

- Start with an initial guess $\beta^{(0)}$.
- Iteratively update your guess as follows.
- At each iteration n, use the Taylor series expansion (up to the quadratic term) around the current value of $\beta^{(n)}$

$$L(\beta) \simeq L(\beta^{(n)}) + L'(\beta^{(n)})(\beta - \beta^{(n)}) + \frac{1}{2}L''(\beta^{(n)})(\beta - \beta^{(n)})^2$$

- Now take the derivative of $L(\beta)$, set it to zero (this would be the likelihood equation for the approximate function), and solve for β.
- Regard the answer as your next guess $\beta^{(n+1)}$:

$$\beta^{(n+1)} = \beta^{(n)} - \frac{L'(\beta^{(n)})}{L''(\beta^{(n)})}$$

- Continue the above process until the algorithm converges.

Newton-Raphson method

- We can rewrite the equation for our next guess as

$$\beta^{(n+1)} = \beta^{(n)} + \frac{u(\beta^{(n)})}{o(\beta^{(n)})}$$

where $u(\beta)$ is the score function, and

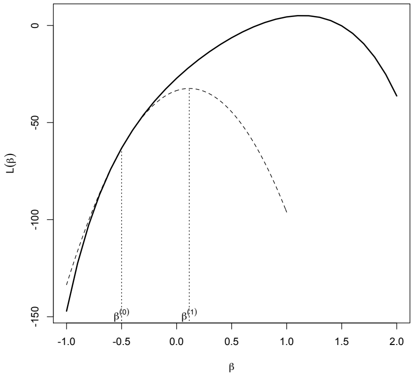
$$o(\beta^{(n)}) = -L''(\beta^{(n)})$$

- $o(\beta)$ is called the *observed information*.
- Note that the Fisher information, $i(\beta) = E[-L''(\beta)]$, is the expected value of the observed information. Unlike the Fisher information, the observed information depends on the the observed data.
- We say the algorithm has converged when

$$|\frac{u(\beta^{(n)})}{o(\beta^{(n)})}| < \epsilon$$

Newton-Raphson method

- The following graph illustrates how this method works.
- The sold line is the log-likelihood function, $\beta^{(0)}$ is our initial guess, the dashed line is the approximate quadratic function around $\beta^{(0)}$, and $\beta^{(1)}$ is our next guess.



Multiple parameter

- For multiple parameter models (where β is a vector), we have

$$\beta^{(n+1)} = \beta^{(n)} + [o(\beta^{(n)})]^{-1} u(\beta^{(n)})$$

- where $o(\beta)$ is a matrix whose (j, k) element is

$$o_{jk}(\beta) = -\frac{\partial^2 L(\beta)}{\partial \beta_j \partial \beta_k}$$

- Therefore, the observed information matrix $o(\beta)$ is the negative of the *Hessian matrix*.
- As before, the expected value of the $o(\beta)$ is the Fisher information matrix.

Fisher scoring algorithm

- If instead of the observed information, we use the expected information, the algorithm is called the *Fisher scoring algorithm*

  $$\beta^{(n+1)} = \beta^{(n)} + [i(\beta^{(n)})]^{-1} u(\beta^{(n)})$$

  where

  $$i_{jk}(\beta) = E[-\frac{\partial^2 L(\beta)}{\partial \beta_j \partial \beta_k}]$$

- It seems that the Fisher scoring algorithm is less sensitive to the initial guess. On the other hand, the Newton-Raphson method tends to converge faster.
- For exponential family models with natural parameters and GLM with canonical links, the two methods are identical (Assignment 2).

Iterative re-weighted least squares

- Fisher scoring is related to the weighted least squares method (e.g., linear regression with non-constant variance for error terms).
- From the above equation for updating β, we have

$$i(\beta^{(n)})\beta^{(n+1)} \;\; = \;\; i(\beta^{(n)})\beta^{(n)} + u(\beta^{(n)})$$

- Recall that for generalized linear models, $i(\beta) = x'wx$. Therefore,

$$(x'w^{(n)}x)\beta^{(n+1)} \;\; = \;\; (x'w^{(n)}x)\beta^{(n)} + u(\beta^{(n)})$$

- After few simple steps, we have

$$(x'w^{(n)}x)\beta^{(n+1)} \;\; = \;\; x'w^{(n)}z^{(n)}$$

where

$$z_i^{(n)} \;\; = \;\; \eta_i^{(n)} + (y_i - \mu_i^{(n)})\frac{\partial \eta_i^{(n)}}{\partial \mu_i^{(n)}}$$

Iterative re-weighted least squares

- At each iteration, we can find the next estimate for β as follows:

$$\beta^{(n+1)} \quad = \quad (x' w^{(n)} x)^{-1} x' w^{(n)} z^{(n)}$$

- The above estimate is similar to the weighted least squares estimate. In this case, $w^{(n)}$ is a diagonal matrix whose i^{th} element is

$$w_{ii}^{(n)} = \frac{\left(\frac{\partial \mu_i^{(n)}}{\partial \eta_i^{(n)}}\right)^2}{var(y_i)}$$

- Note that for GLM, the weights, w, and the response variable, z, changes from one iteration to another based on the current estimate of β.
- We iteratively estimate β until the algorithm converges.

MLE using R

- You can of course use R to estimate the parameter of generalized linear models.
- The function has the following format:

```
glm( formula, family, data )
```

- Assignment 2: Write your own code for estimating the parameters of logistic regression models using 1) Newton-Raphson method, 2) iterative re-weighted least squares method. Try your program on the Pima data set (from `MASS` package in R), where the objective is to investigate factors involved in diabetes among women of Pima Indian heritage. Model the response variable `type` based on the `Pima.tr` data set. Use your estimates to predict the response variable `type` in the `Pima.te` data set. Evaluate your model based on its prediction accuracy and ROC curve.

Logistic regression model

- Recall that for binary response variable, we use Bernoulli distribution (or Binomial if $n_i > 1$) for the random component:

$$y_i|\theta_i \sim \mathrm{Bernoulli}(\mu_i)$$

- In this case, a common link function to connect the random component to the systematic component $\eta_i = x_i\beta$ is the *logit* function defined as follows:

$$g(\mu_i) = \log(\frac{\mu_i}{1-\mu_i}) = \log[\frac{P(y_i = 1|x_i, \beta)}{1 - P(y_i = 1, \beta|x_i)}] = x_i\beta$$

- For this model,

$$\mu_i \quad = \quad P(y_i = 1|x_i, \beta) = \frac{\exp(x_i\beta)}{1 + \exp(x_i\beta)}$$

- In this model, $\beta = (\beta_0, \beta_1, ..., \beta_p)$

Logistic regression model

- The likelihood function for the general case where $n_i > 1$ is defined in terms of β as follows:

$$\begin{aligned} p(y|\mu) &\propto \prod_{i=1}^{n} \mu_i^{y_i}(1-\mu_i)^{n_i-y_i} \\ p(y|\beta) &\propto \prod_{i=1}^{n} \left(\frac{\exp(x_i\beta)}{1+\exp(x_i\beta)}\right)^{y_i} \left(\frac{1}{1+\exp(x_i\beta)}\right)^{n_i-y_i} \end{aligned}$$

- Recall that the score function for this model is as follows:

$$u_j(\beta) = \sum_{i=1}^{n} [y_i - n_i \frac{\exp(x_{i\beta})}{1+\exp(x_i\beta)}] x_{ij}$$

  where $u(\beta)$ is a $p+1$ vector.

- The Fisher information is

$$i_{jk}(\beta) = \sum_{i=1}^{n} n_i x_{ij} x_{ik} \frac{\exp(x_{i\beta})}{[1+\exp(x_i\beta)]^2}$$

Maximum likelihood estimation

- To find MLE of β, we showed that we could either use Newton-Raphson (which is the same as Fisher scoring algorithm in this case) or iterative re-weighted least squares.
- As usual, asymptoticly, $cov(\hat{\beta}) = [i(\hat{\beta})]^{-1}$.
- The standard error for each β is obtained by taking the square root of the corresponding diagonal element of $cov(\hat{\beta})$.
- To interpret β, notice that $\log[\frac{P(y_i=1|x_i,\beta)}{1-P(y_i=1,\beta|x_i)}]$ is the log of odds for the outcome of interest, $y_i = 1$.
- The intercept β_0 is therefore the log of odds when the value of all covariates is 0.
- Or we can say, $\exp(\beta_0)$ is the odds when all covariates are 0.

Maximum likelihood estimation

- $\exp(\beta_j)$ on the other hand is how much the odds multiplicatively increases for one unit increase in x_j when all other covariates are fixed.
- Or we can say, $\exp(\beta_j)$ is the odds ratio for subjects with $X_j = x_j + 1$ compared to subjects with $X_j = x_j$ when all other covariates are fixed.
- Positive β_j indicates that the odds increases as x_j increases (everything else fixed), where is for negative estimate of β_j the odds decreases as x_j increases (everything else fixed).

Inference

- To decide whether β_j is statistically significant or not, as usual, we can use one of the three likelihood based tests.
- Using the asymptotic normality of $\hat{\beta}$, we can use the Wald test to make inference about the significance β's.
- The [univariate] test statistic is

$$z = \frac{\hat{\beta}_j}{se(\hat{\beta}_j)}$$

where z^2 has an asymptotic null distribution χ^2_1.

Inference

- Alternatively, we can use the score test with the following test statistic

  $$s^2 = \frac{u(\beta_j = 0)}{i_{jj}(\beta_j = 0)}$$

  where s^2 has an asymptotic null distribution χ^2_1.

Inference

- In most cases, the better test would be the likelihood ratio test,

$$\Lambda = -2(L_1 - L_0)$$

  where L_1 is the maximum log-likelihood for the unrestricted model with $\beta_j \neq 0$, and L_0 is the maximum log-likelihood for the restricted model with $\beta_j = 0$.
- Λ has an asymptotic null distribution χ^2_1.

Model selection for inference about relationships

- Similar to ordinary linear regression analysis, modeling binary response variables involves many decisions regarding the type of model.
- The process of model selection in general includes a series of decisions about the systematic component and (also the type of link function).
- For the systematic component, we need to start with a list of potential covariates. This by itself is a subjective decision. That is, we exclude many variables that in or opinion are not related to the response variable.
- From the list of potential covariates, we need to choose the ones that are related to the response variable based the evidence provided by the data.

Model selection for inference about relationships

- A common procedure (again similar to ordinary regression analysis) is the stepwise model selection procedure. Where we add variables that are significant and remove the ones that are not significant (given all other variables already in the model).
- Some ther model selection criteria are *Akaike information criterion* (AIC), and *Bayesian information criterion* (BIC).
- AIC penalizes (i.e., increases) the deviance (-2 times of log-likelihood) by $2df$, whereas BIC penalizes the deviance by $\log(n)df$, where n is the sample size.
- In R, you can use `stepAIC` in the package MASS to perform stepwise selection using either AIC or BIC.

Model selection for prediction

- If our objective for building a logistic regression model is to predict the values of response variable for future observations, it makes more sense to select the model that would help us in prediction.
- For this purpose, we could build the model on one part of the data, called the *training set*, fine-tune it on another part, called the *validation set*, and testing on the third part, called the *test set*.
- Alternatively, we could use *cross-validation* or *leave-one-out* procedure.
- When apply our model to the test set, we need a good measurement for evaluating the predictive power of the model; that is, how well our model can identify the correct class (0 or 1) for future observations.
- We will discuss some of these measurements next.

Predictive power

- A common measure for predictive power is *accuracy rate*, which is defined as the percentage of the times the correct class (0 or 1 in this case) is predicted for future observations (or observations in the test set).

$$acc \quad = \quad \frac{\sum_{i=1}^{n_t} I(\hat{y}_i = c_i)}{n_t}$$

  where n_t is the number of observations in the test set, c_i is the true class, and $\hat{y}_i$ is the predicted class for i^{th} observation in the test set. The index i here is for test cases.

- Instead of accuracy rate, we could also use error rate, which is defined as the percentage of the times the wrong class is predicted.

Predictive power

- Note that the outputs of logistic regression models are in fact between 0 and 1, which are interpreted as probabilities.
- Therefore, we need to set an appropriate cutoff to obtain $\hat{y}$ as a binary prediction.
- In general, the cutoff depends on the loss function; that is, the cost of predicting the class as 0, when the true class is 1, and vice versa.
- In most practical problems, the costs of misclassifying 0 as 1 and 1 as 0 are not the same.
- For 0-1 loss function, we assign a test case to the class with the highest probability; that is, we set the cutoff at 0.5.

Predictive power

- Instead of averaging over all predictions, it might be more informative to separate the types of error.
- One common approach for doing this is to present the results in a *classification table* (a.k.a, *confusion matrix*)

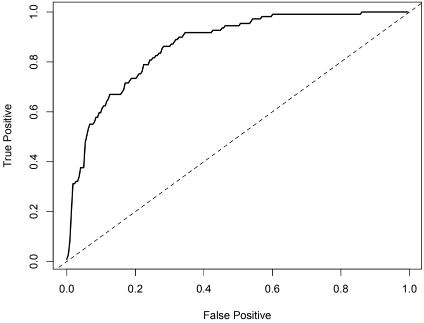
		Predicted class	
		0	1
True class	0	True Negative	False Positive
	1	False Negative	True Positive

- Based on this table, we have

$$\begin{aligned} \text{Sensitivity} &= P(\hat{y} = 1 | y = 1) \\ \text{Specificity} &= P(\hat{y} = 0 | y = 0) \end{aligned}$$

ROC

- Receiver Operating Characteristic (ROC) curve allows for simultaneous consideration of sensitivity and specificity without setting an arbitrary cut-off.
- The curve plots sensitivity (true positive) as a function of 1-specificity (false positive).



ROC

- Each point on the curve corresponds to a specific value of the cutoff.
- A more accurate model will have an ROC curve further away from the diagonal line (random model) with perfect prediction corresponding to the (0, 1) point.
- The Area Under the ROC Curve (AUC) is used as a summary statistic to compare models. For a perfect model, the AUC is equal to 100%.

Predictive power

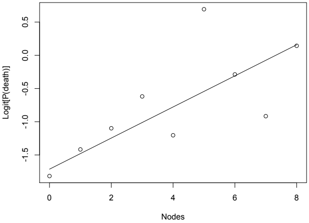
- Instead of setting a cutoff, we can also use *average log-probability* (based on the estimated probability of the correct class) as a measure of prediction accuracy.

  $$\frac{\sum_{i=1}^{n_t} \log[P(y_i = c_i|x_i)]}{n}$$

  where c_i is the correct class for the future observation (x_i, c_i), or the observation in the test set.

Deciding on whether to use logistic model

- For simple cases with only few covariates, we might be able to find out whether fitting a logistic model is a good idea.
- For example, we can plot the logit function vs. each covariate to make sure the relationship is close to linear. The following plot shows the logit function of death due to breast cancer vs. the number of tumor nodes.



- For continuous variables, we could first discretize the variable so the proportions are not 0 or 1.

Deciding on whether to use logistic model

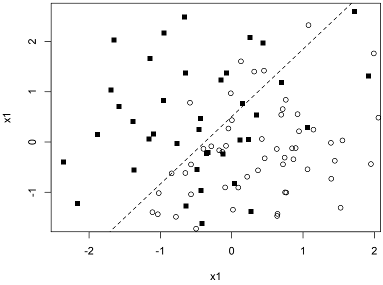
- If we look at logistic regression as a classifier, we realize the decision boundary is a hyperplane since the boundary is where $P(y = 1|x, \beta) = P(y = 0|x, \beta)$.
- Therefore, at the boundary we have

$$log(\frac{P(y = 1|x, \beta)}{1 - P(y = 1|x, \beta)}) = x\beta = 0$$

- Where $\{x|x\beta = 0\}$ is a hyperplane.

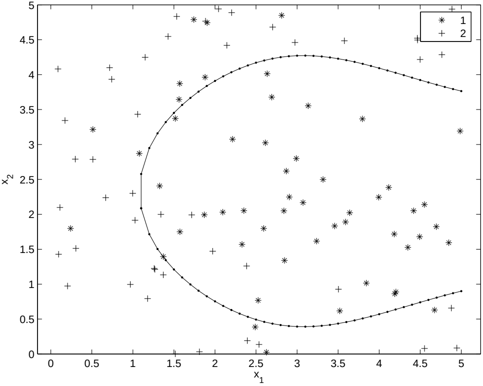
Deciding on whether to use logistic model

- For two dimensional covariates, the above hyperplane is of course a straight line.
- Therefore, logistic regression as a classifier use a hyperplane (e.g., straight line in $\mathcal{R}^2$) to separate the two classes.
- We could plot the data to see whether a straight line boundary is suitable.



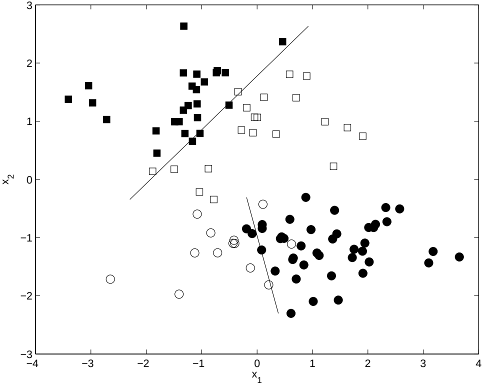
Deciding on whether to use logistic model

- If the decision boundary does not seem to be linear, logistic regression might not do well.



Deciding on whether to use logistic model

- Sometimes, we could fix this by using a mixture of logistic regression models



Diagnosis

- For diagnosis, we usually use deviance residuals

$$dr_i = sign(y_i - n_i\hat{\theta}_i)\sqrt{d_i}$$

where

$$d_i = 2\sum_i \{y_i \log(\frac{y_i}{n_i\hat{\theta}}) + (n_i - y_i)\log(\frac{n_i - y_i}{n_i - n_i\hat{\theta}})\}$$

- Or we can use Pearson's residuals

$$e_i = \frac{y_i - n_i\hat{\theta}_i}{\sqrt{n_i\hat{\theta}_i(1-\hat{\theta}_i)}}$$

- Or standardized Pearson's residuals

$$r_i = \frac{e_i}{\sqrt{1-h_i}}$$

Noncanonical link- Probit

- The logit link we have been using so far is the inverse CDF of logistic distribution (with mean $\mu = 0$ and scale $s = 1$).
- Obviously, we could use the inverse CDF of any continuous distribution to map real numbers (η in this case) to $[0, 1]$ interval.
- One possibility, which is the most popular after inverse CDF of logistic distribution, is to use the inverse CDF of standard normal distribution, Φ^{-1}.

$$\Phi^{-1}(\mu_i) \quad = \quad \eta_i$$

Noncanonical link- Probit

- Recall that for non-canonical link, the score function is

$$u(\beta_j) = \sum_i \frac{(y_i - \mu_i)}{var(y_i)} x_{ij} \frac{\partial \mu_i}{\partial \eta_i}$$

- Therefore, for probit model

$$u(\beta_j) = \sum_i \frac{(y_i - n_i\theta_i)}{n_i\theta_i(1 - \theta_i)} x_{ij}\phi(\eta_i)$$

  where ϕ is the density function for the standard normal distribution.

- Note that it is more common to use the logit link since the estimates can be interpreted in terms of odds.

Noncanonical link- Complementary log-log link

- Both logit and probit models are symmetric around $\theta = 0.5$; $g(\theta) = -g(1-\theta)$.
- Therefore, the underlying assumption for using these models is that θ approaches 0 and 1 with the same rate.
- If this is not appropriate, we can use another link function called *complementary log-log*, which is not symmetric

$$g(\theta_i) = \log[-\log(1-\theta_i)] = \eta_i$$

- This function approaches 1 more sharply than it approaches 0.
- If we want the function approaches 0 faster, we can use the *log-log* link, which is $g(\theta) = \log(-\log(\theta))$, or switch 0 and 1.
- For the log-log link, as x increases the the function is monotone decreasing when $\beta > 0$, and monotone increasing when $\beta < 0$.

Noncanonical link- Complementary log-log link

- In this model, for two possible values of x_j, denoted as x_{j1} and x_{j2}, we have

$$[1 - P(y = 1|x_{j2})] \quad = \quad [1 - P(y = 1|x_{j1})]^{\exp[(x_{j2} - x_{j1})\beta_j]}$$

- Therefore, the interpretation of β_j for this model is: for one unit increase in x_j (everything else fixed), the complement probability $(1 - \theta)$ raises to the power $\exp(\beta_j)$