Lecture 8: Logistic Regression

Statistical Learning and Data Mining

Xingye Qiao

Department of Mathematical Sciences
Binghamton University

E-mail: qiao@math.binghamton.edu

Read: ISLR Chs. 4.1–4.3, ESLII Chs. 4.1 and 4.4, SLS Chs. 3.1–3.2

Outline

Why not perform regression on indicators?

- Suppose we try to predict medical condition of a patient in ER on the basis of her symptoms.
- 3 possible diagnoses: stroke, drug overdose, epileptic seizure.
- Coding: $1 =$ stroke, $2 =$ drug overdose, $3 =$ epileptic seizure.
- This implies an ordering between the three conditions.
- If coding is changed, the resulting linear model will be fundamentally different.
- 1, 2, 3 coding will be reasonable only if
  - label's values take natural ordering
  - gaps between adjacent labels are similar
- No quantitative ways to verify these.

- The situation is better for binary classification: $Y = 0$ or 1.
- Linear regression actually tries to estimate

$$\mathrm{E}(Y|X) = P(Y = 1|X)$$

- However, some fitted Y values may be out of $(0, 1)$
- Such dummy variable approach does not make sense for categorical response with more than two levels.

The next section would be ......

Logit function

- Goal: to model $p(x) := P(Y = 1|X = x)$
- Linear regression would model it as $p(x) = \beta_0 + x^T \boldsymbol{\beta}_1$.
- Not a good idea since $p(x)$ should $\in (0, 1)$
- Need a function: maps $p(x)$ to $\mathbb{R}$ then modelled by $\beta_0 + x^T \boldsymbol{\beta}_1$. We may use the logit function

$$\operatorname{logit}(p(x)) = \log\left(\frac{p(x)}{1 - p(x)}\right)$$

- Model:

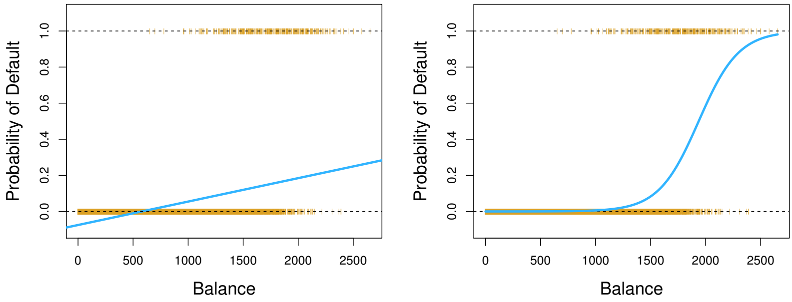
$$\log\left(\frac{p(x)}{1 - p(x)}\right) = \beta_0 + x^T \boldsymbol{\beta}_1$$

$$\text{odds: } \frac{p(x)}{1 - p(x)} = e^{\beta_0 + x^T \boldsymbol{\beta}_1}$$

Univariate Logistic Regression.

For simplicity, consider univariate case (only one predictor)

$$p(x) = \frac{e^{\beta_0 + x^T \beta_1}}{1 + e^{\beta_0 + x^T \beta_1}}$$



The next section would be ......

- Logistic regression is a special case of generalized linear models (McCullagh and Nelder 1989).

- These models describe the response variable using a member of the exponential family, which includes the Bernoulli, Poisson, and Gaussian as particular cases.

- A transformed version of the response mean $\mathrm{E}[Y|X = x]$ is then approximated by a linear model.

Intro to GLM

A generalized linear model (GLM) generalizes normal linear regression models to address a broader class of data structures.

- instead of being normal, the response could have any distribution from the exponential family.
- instead of identity, other function (called link function) can map $\mu_i = \mathrm{E}(Y|X = x_i)$ to $\mathbb{R}$ and then be modelled by a linear function.

$$g(\mu_i) = \eta := \beta_0 + x_i^T \boldsymbol{\beta}$$

In linear regression, $g(\mu) = \mu$.
In logistic regression, $g(\mu) = \log(\mu/(1-\mu))$

Exponential family

Y from an exponential family has a density with the following form,

$$f_Y(y; \theta, \phi) = e^{\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)}$$

for specific functions $a(\cdot)$, $b(\cdot)$ and $c(\cdot)$.

- Example: Gaussian distribution. $\theta = \mu$, $\phi = \sigma^2$, $a(\phi) = \phi$, $b(\theta) = \frac{\theta^2}{2}$, $c(y, \phi) = -\frac{1}{2}(\frac{y^2}{\phi} + \log(2\pi\phi))$

- Example: Binomial distribution:

$$
\begin{aligned}
&\binom{m}{y}(\mu/m)^y(1-\mu/m)^{m-y} \\
=&\binom{m}{y}p^y(1-p)^{m-y} \\
=&\exp[y\log p+(m-y)\log(1-p)+\log\binom{m}{y}] \\
=&\exp[y\log(p/(1-p))+m\log(1-p)+\log\binom{m}{y}]
\end{aligned}
$$

Hence $\theta = \log(p/(1-p))$,
$b(\theta) = -m\log(1-p) = m\log(1+e^{\theta})$

Link function

$$f_Y(y;\theta,\phi) = e^{\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)}$$

Three interrelated parameters:

1. θ: canonical parameter
2. $\mu := \mathrm{E}Y$: mean parameter, which can be shown to be $b'(\theta)$
3. $\eta := X^T\beta$.

A link function g, to be determined, relates the linear predictor η to the mean parameter μ.

$$\eta = g(\mu)$$

A canonical link occurs when $\eta = \theta$. $\Rightarrow (b')^{-1}(\mu) = \theta = \eta = g(\mu)$

Table 2.1 *Characteristics of some common univariate distributions in the exponential family*†

	Normal	Poisson	Binomial	Gamma	Inverse Gaussian
Notation	$N(\mu,\sigma^2)$	$P(\mu)$	$B(m,\pi)/m$	$G(\mu,\nu)$	$IG(\mu,\sigma^2)$
Range of y	$(-\infty,\infty)$	$0(1)\infty$	$\frac{0(1)m}{m}$	$(0,\infty)$	$(0,\infty)$
Dispersion parameter: ϕ	$\phi=\sigma^2$	1	$1/m$	$\phi=\nu^{-1}$	$\phi=\sigma^2$
Cumulant function: $b(\theta)$	$\theta^2/2$	$\exp(\theta)$	$\log(1+e^\theta)$	$-\log(-\theta)$	$-(-2\theta)^{1/2}$
$c(y;\phi)$	$-\frac{1}{2}\left(\frac{y^2}{\phi}+\log(2\pi\phi)\right)$	$-\log y!$	$\log\binom{m}{my}$	$\nu\log(\nu y)-\log y$ $-\log\Gamma(\nu)$	$-\frac{1}{2}\left\{\log(2\pi\phi y^3)+\frac{1}{\phi y}\right\}$
$\mu(\theta)=E(Y;\theta)$	θ	$\exp(\theta)$	$e^\theta/(1+e^\theta)$	$-1/\theta$	$(-2\theta)^{-1/2}$
Canonical link: $\theta(\mu)$	identity	log	logit	reciprocal	$1/\mu^2$
Variance function: $V(\mu)$	1	μ	$\mu(1-\mu)$	μ^2	μ^3

†The mean-value parameter is denoted by μ, or by π for the binomial distribution.
The parameterization of the gamma distribution is such that its variance is μ^2/ν.
The canonical parameter, denoted by θ, is defined by (2.4). The relationship between μ and θ is given in lines 6 and 7 of the Table.

Deviance

- Use canonical link, then $g(\mu) = (b')^{-1}(\mu)$ and $\theta = g(\mu)$
- Define log-likelihood: $\ell(\theta(\mu); y, \phi) = \frac{y\theta - b(\theta)}{a(\phi)} + c(y, \phi)$
- Compare two models:
  - Saturated model S: each μ_i has its own set of β_i
  - Model M: μ_i share the same β
- The deviance is twice the difference between the log-likelihood of these two models.

$$2 \times [\, \underbrace{\ell(\theta(y))}_{\text{model } S} - \underbrace{\ell(\theta(\hat{\mu}))}_{\text{model } M} \,]$$

- Intuitively, this measures a goodness-of-fit. Note that the saturated model S is perfect in fitting.
- For Gaussian distribution, boil down to squared error.

Deviance for Binomial data

- $Y_i \sim Bin(p, m)$ - Bernoulli is special case with $m = 1$.
- $b(\theta) = m \log(1 + e^\theta)$
- $b'(\theta) = m \dfrac{e^\theta}{1 + e^\theta}$
- Canonical link: $g(\mu) = (b')^{-1}(\mu) = \log(\frac{\mu/m}{1-\mu/m}) = \log(\frac{p}{1-p})$; note $p = \mu/m$
- Estimate to p under S: y/m
- Estimate to p under M: $\hat{p}$
- Log-likelihood: $y \log p + (m - y) \log(1 - p) + \log \binom{m}{y}$
- Deviance (similar to RSS in linear regression):

$$
\begin{aligned}
&2\{[y \log(y/m) + (m - y) \log(1 - y/m)] - [y \log \hat{p} + (m - y) \log(1 - \hat{p})\\
=&2\{y \log(\frac{y/m}{\hat{p}}) + (m - y) \log(\frac{1 - y/m}{1 - \hat{p}})\}
\end{aligned}
$$

The next section would be ......

Fitting logistic regression

- We introduce two algorithms of fitting logistic regression.
- They are introduced here not because you are expected to re-implement logistic regression, but because they may be useful for developing other statistical learning algorithms.
  1. Newton-Raphson.
  2. Coordinate descent.

Conditional likelihood

- Data: $\{(\boldsymbol{x}_i, y_i), i = 1, \ldots, n\}$ and $y_i = 0, 1$
- Given $\boldsymbol{X}_i = \boldsymbol{x}_i$, Y_i is a Bernoulli random variable (conditionally) with success probability $p(\boldsymbol{x})$.
- Let $f(\boldsymbol{x}) = \beta_0 + \boldsymbol{x}^T \boldsymbol{\beta}$
- Conditional likelihood of $(\beta_0, \boldsymbol{\beta})$ is

$$\prod_{i=1}^{n} p(\boldsymbol{x}_i; \beta_0, \boldsymbol{\beta})^{y_i} [1 - p(\boldsymbol{x}_i; \beta_0, \boldsymbol{\beta})]^{1-y_i}$$

- Conditional log-likelihood of $(\beta_0, \boldsymbol{\beta})$ is

$$\begin{aligned}
\ell(\beta_0, \boldsymbol{\beta}) &:= \sum_{i=1}^{n} \{y_i \log(p(\boldsymbol{x}_i; \beta_0, \boldsymbol{\beta})) + (1 - y_i) \log[1 - p(\boldsymbol{x}_i; \beta_0, \boldsymbol{\beta})]\} \\
&= \sum_{i=1}^{n} \left\{ y_i \log \left[\frac{\exp(f(\boldsymbol{x}_i))}{1 + \exp(f(\boldsymbol{x}_i))} \right] + (1 - y_i) \log \left[\frac{1}{1 + \exp(f(\boldsymbol{x}_i))} \right] \right\}
\end{aligned}$$

$$
\begin{aligned}
\ell(\beta_0, \boldsymbol{\beta}) &:= \sum_{i=1}^{n} \{y_i \log(p(\boldsymbol{x}_i; \beta_0, \boldsymbol{\beta})) + (1 - y_i) \log[1 - p(\boldsymbol{x}_i; \beta_0, \boldsymbol{\beta})]\} \\
&= \sum_{i=1}^{n} \left\{ y_i \log \left[\frac{\exp(f(\boldsymbol{x}_i))}{1 + \exp(f(\boldsymbol{x}_i))} \right] + (1 - y_i) \log \left[\frac{1}{1 + \exp(f(\boldsymbol{x}_i))} \right] \right\} \\
&= \sum_{i=1}^{n} \{y_i f(\boldsymbol{x}_i) - \log[1 + \exp(f(\boldsymbol{x}_i))]\} \\
&= \sum_{i=1}^{n} \{y_i(\beta_0 + \boldsymbol{\beta}' \boldsymbol{x}_i) - \log[1 + \exp(\beta_0 + \boldsymbol{\beta}' \boldsymbol{x}_i)]\}
\end{aligned}
$$

The maximizer of $\ell(\beta_0, \boldsymbol{\beta})$, say $(\beta_0^*, \boldsymbol{\beta}^*)$, can be plugged into $f(\boldsymbol{x}; \beta_0, \boldsymbol{\beta})$

$$
f(\boldsymbol{x}) = \beta_0^* + \boldsymbol{x}^T \boldsymbol{\beta}^*
$$

1. $f(\boldsymbol{x}) > 0 \Rightarrow p(\boldsymbol{x}) > 1/2 \Rightarrow Y$ is more likely to be 1
2. $f(\boldsymbol{x}) < 0 \Rightarrow p(\boldsymbol{x}) < 1/2 \Rightarrow Y$ is more likely to be 0

Optimization

For simplicity, view $(\beta_0, \boldsymbol{\beta}')'$ as new $\boldsymbol{\beta}$ and $(1, \boldsymbol{x}')'$ as new $\boldsymbol{x}$
Search solution $\boldsymbol{\beta}$ to score equation

$$\dot{\ell}(\boldsymbol{\beta}) = \mathbf{0}$$

- Recall univariate Newton-Raphson method: find root of $f(x) = 0$. Iteratively do:

$$x_{n+1} \leftarrow x_n - f(x_n)/f'(x_n)$$

  Motivated by Taylor expansion.
- Here:

$$\boldsymbol{\beta}^{(k+1)} \leftarrow \boldsymbol{\beta}^{(k)} - [\ddot{\ell}(\boldsymbol{\beta}^{(k)})]^{-1}\dot{\ell}(\boldsymbol{\beta}^{(k)}),$$

  where $\ddot{\ell}(\boldsymbol{\beta})$ is the Hessian matrix, i.e. $(\ddot{\ell}(\boldsymbol{\beta}))_{ij} = \partial_i\partial_j\ell(\boldsymbol{\beta})$

Calculations lead to

$$
\begin{aligned}
\dot{\ell}(\boldsymbol{\beta}) &= \sum_{i=1}^{n} \boldsymbol{x}_i\{y_i - p(\boldsymbol{x}_i;\boldsymbol{\beta})\} = \mathbf{X}(\boldsymbol{y} - \boldsymbol{p}) && (1)\\
&\text{where } \boldsymbol{p} := (p(\boldsymbol{x}_1;\boldsymbol{\beta}), \ldots, p(\boldsymbol{x}_n;\boldsymbol{\beta}))^T && (2)\\
\ddot{\ell}(\boldsymbol{\beta}) &= \frac{\partial}{\partial \boldsymbol{\beta}^T}\mathbf{X}(\boldsymbol{y} - \boldsymbol{p}) && (3)\\
&= -\mathbf{X}\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{\beta}^T} && (4)\\
&= -\mathbf{X}\mathbf{W}\mathbf{X}^T && (5)
\end{aligned}
$$

where $\mathbf{W} = \mathrm{Diag}\{p(\boldsymbol{x}_i;\boldsymbol{\beta})[1 - p(\boldsymbol{x}_i;\boldsymbol{\beta})]\}$.
Note $\frac{\partial p(\boldsymbol{x}_i;\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = p(\boldsymbol{x}_i;\boldsymbol{\beta})[1 - p(\boldsymbol{x}_i;\boldsymbol{\beta})]\boldsymbol{x}_i$ in the last step,
so $\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{\beta}^T} = \mathbf{W}\mathbf{X}^T$.

Write the N-R method as

$$\begin{aligned}
\boldsymbol{\beta}^{(k+1)} &\leftarrow \boldsymbol{\beta}^{(k)} - [\ddot{\ell}(\boldsymbol{\beta}^{(k)})]^{-1}\dot{\ell}(\boldsymbol{\beta}^{(k)}) && (6)\\
&= \boldsymbol{\beta}^{(k)} + [\mathbf{X}\mathbf{W}\mathbf{X}^T]^{-1}\mathbf{X}(\boldsymbol{y} - \boldsymbol{p}) && (7)\\
&= [\mathbf{X}\mathbf{W}\mathbf{X}^T]^{-1}\mathbf{X}\mathbf{W}[\mathbf{X}^T\boldsymbol{\beta}^{(k)} + \mathbf{W}^{-1}(\boldsymbol{y} - \boldsymbol{p})] && (8)\\
&= [\mathbf{X}\mathbf{W}\mathbf{X}^T]^{-1}\mathbf{X}\mathbf{W}\boldsymbol{z} && (9)
\end{aligned}$$

- This is exactly the same as the solution to weighted least square with design matrix $\mathbf{X}$, response variable $\boldsymbol{z}$ and weights $p(\boldsymbol{x}_i;\boldsymbol{\beta})[1 - p(\boldsymbol{x}_i;\boldsymbol{\beta})]$.
- One must update the response variable $\boldsymbol{z}$ and the weight matrix $\mathbf{W}$ for each iteration.
- Convergence is NOT guaranteed. $\mathbf{W}$ and $\mathbf{X}\mathbf{W}\mathbf{X}^T$ must be invertible.
- Data separation issue: if two classes are well separated, all $p(\boldsymbol{x}_i)$ are too close to 0 or 1 $\Rightarrow$ $\mathbf{W}$ is almost $\mathbf{0}$ (trouble!)

Deviance Loss function.

- The deviance $2\{y\log(\frac{y/m}{\hat{p}}) + (m-y)\log(\frac{1-y/m}{1-\hat{p}})\}$ motivates a natural loss function for logistic regression.
- Like minimizing RSS in linear regression, we minimize the deviance in logistic regression, i.e. (assume no group data $m_i = 1$)

$$\min \sum_{i=1}^{n} y_i \log(\frac{y_i}{\hat{p}}) + (1-y_i)\log(\frac{1-y_i}{1-\hat{p}})$$

- Delete term irrelevant to $\hat{p}$:

$$\min \sum_{i=1}^{n} \{y_i \log(\frac{1-\hat{p}}{\hat{p}}) - \log(1-\hat{p})\} = \{-y_i f(\boldsymbol{x}_i) + \log(1+e^{f(\boldsymbol{x}_i)})\}$$

- This is equivalent to the conditional likelihood derived earlier.

Alternative coding for logistic regression

Recall for $y_i = 0,\ 1$

$$y_i \log\left(\frac{\exp(f(\boldsymbol{x}_i))}{1+\exp(f(\boldsymbol{x}_i))}\right) + (1-y_i)\log\left[\frac{1}{1+\exp(f(\boldsymbol{x}_i))}\right]$$

This is equivalent to the following function for coding $y_i = \pm 1$

$$\log\left(\frac{1}{\exp(-y_i f(\boldsymbol{x}_i))+1}\right) = -\log\left(\exp(-y_i f(\boldsymbol{x}_i))+1\right)$$

Logistic regression can be viewed as **minimizing** over $(\boldsymbol{\beta}, \beta_0)$

$$\sum_{i=1}^{n} \log\left(\exp(-y_i f(\boldsymbol{x}_i))+1\right) = \sum_{i=1}^{n} L(y_i f(\boldsymbol{x}_i))$$

where the loss function

$$L(u) = \log(\exp(-u)+1)$$

and $f(\boldsymbol{x}) = \boldsymbol{\beta}^T \boldsymbol{x} + \beta_0$

Gradient descent optimization

The gradient descent algorithm takes the following update iteratively to minimize $f(\boldsymbol{\omega})$:

$$\boldsymbol{\omega}_{(k+1)} \leftarrow \boldsymbol{\omega}_{(k)} - \gamma f'(\boldsymbol{\omega}_{(k)})$$

where $0 < \gamma \leq 1$ is step size (or learning rate.)

- Compared to the Newton-Raphson method

  $$\boldsymbol{\omega}_{(k+1)} \leftarrow \boldsymbol{\omega}_{(k)} - (f''(\boldsymbol{\omega}))^{-1} f'(\boldsymbol{\omega}_{(k)}),$$

  the gradient descent method directly updates the point toward the direction of steepest descent for f, while Newton-Raphson method essentially indirectly optimizes by finding the root of $f'(\boldsymbol{\omega}) = 0$
- The direction $\gamma f'(\boldsymbol{\omega}_{(k)})$ is different from $(f''(\boldsymbol{\omega}))^{-1} f'(\boldsymbol{\omega}_{(k)})$.
- N-R should converge sooner than gradient descent. The latter may call for many iterations.

Again, let $(1, \boldsymbol{x}')'$ be viewed as new $\boldsymbol{x}$. The goal is to minimize over $\boldsymbol{\omega} = (\beta_0, \boldsymbol{\beta})$

$$f(\omega) := \sum_{i=1}^{n} \log[1 + \exp(-y_i \boldsymbol{\omega}^T \boldsymbol{x}_i)]$$

whose gradient is

$$\begin{aligned} f'(\omega) &:= \sum_{i=1}^{n} \frac{-\exp(-y_i \boldsymbol{\omega}^T \boldsymbol{x}_i)}{1 + \exp(-y_i \boldsymbol{\omega}^T \boldsymbol{x}_i)} y_i \boldsymbol{x}_i \\ &= \sum_{i=1}^{n} \left\{ \frac{1}{1 + \exp(-y_i \boldsymbol{\omega}^T \boldsymbol{x}_i)} - 1 \right\} y_i \boldsymbol{x}_i \end{aligned}$$

At each iteration, we calculate the gradient, and then update according to

$$\boldsymbol{\omega}_{(k+1)} \leftarrow \boldsymbol{\omega}_{(k)} - \gamma f'(\boldsymbol{\omega}_{(k)})$$

The next section would be . . . . . .

Interpretation of the result

- $\dfrac{p(x)}{1-p(x)} = e^{\beta_0+\beta_1 x_1+\cdots+\beta_j x_j+\cdots+\beta_p x_p}$
- If x_j is increased to x_j+1
  $\dfrac{\tilde{p}(x)}{1-\tilde{p}(x)} = e^{\beta_0+\beta_1 x_1+\cdots+\beta_j x_j+\beta_j\cdot 1+\cdots+\beta_p x_p} = e^{\beta_j} \times \dfrac{p(x)}{1-p(x)}$
- Every time variable x_j is increased by 1 unit, the odd of the event is multiplied by a factor of e^{β_j}. Note that e^{β_j} may be less than 1.
- The effect on the probability p itself is more complicated. However, note that

  $$\partial_{x_j} p = p(1-p)\beta_j$$

  So the effect is large when p is near 0.5 than when p is close to 0 or 1.

Bias and precision of estimates

- For large n,

$$\begin{aligned} \mathrm{E}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) &= O(n^{-1}) \\ \mathrm{Var}(\hat{\boldsymbol{\beta}}) &= (\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\{1 + O(n^{-1})\} \end{aligned}$$

Hypothesis test: LRT

- Likelihood ratio test.
- Compare two nested model
- The difference between the deviances for the bigger and smaller models is asymptotically χ^2 distributed under the smaller model.
- Use the function `anova()`
- Recall ANOVA in Linear Regression.