Chapter 15

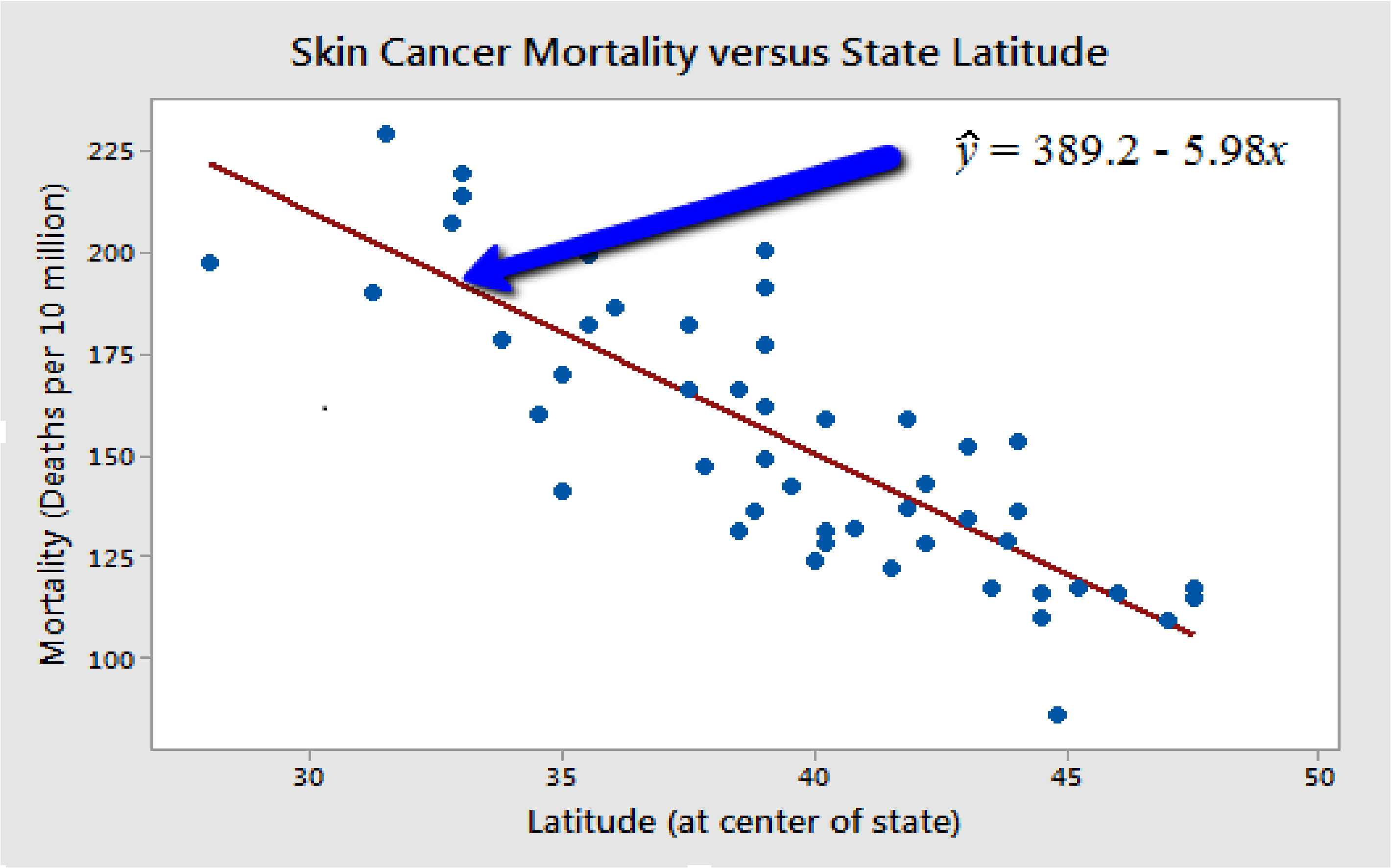
Generalized Linear Model

Chanwoo Yoo, Division of Advanced Engineering,
Korea National Open University

Contents

1. Generalized Linear Model

1. Simple Linear Regression Revisited



2. Comparison of GLMs

	Linear Regression	Logistic Regression	Poisson Regression
Response	Continous	Binary	Count
$Y\|X=x$	Normal	Bernoulli	Poisson
Parameter	μ,σ^2	π	λ
g^{-1}	$\mu = X\beta$	$\pi = \dfrac{exp(X\beta)}{1+exp(X\beta)}$	$\lambda = exp(X\beta)$
Link Function g	Identity	$\log\left(\dfrac{\pi}{1-\pi}\right)$	$\log(\lambda)$

3. Generalized Linear Model (GLM)

- Generalized linear models provides a generalization of ordinary least squares (OLS) regression that relates the random term (the response Y) to the systematic term (the linear predictor $X\beta$) via a link function (denoted by $g(\cdot)$).

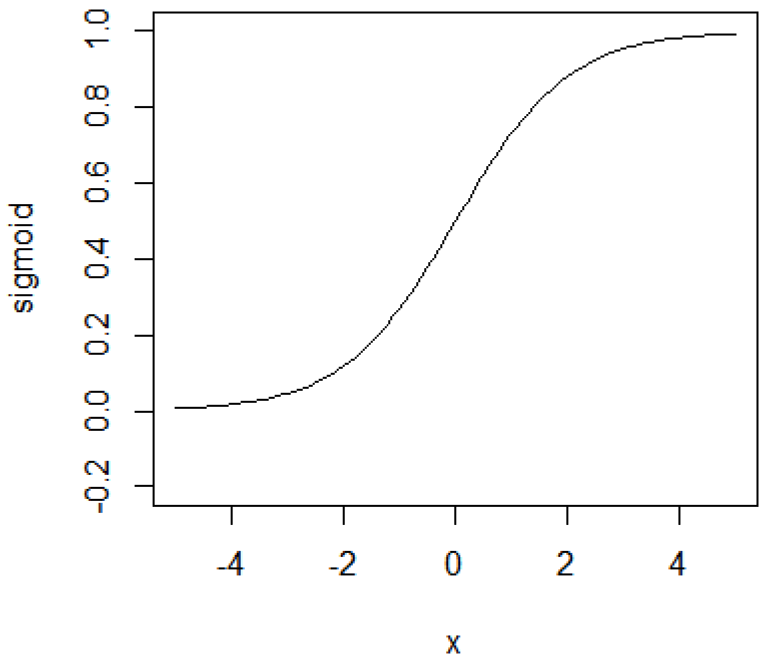
2. Logistic Regression

1. Binary Logistic Regression

- Pros
  - Interpretable
  - Efficient in terms of time and space
  - Foundation of deep learning
- Cons
  - Does not perform as well as more complex methods like gradient boosted trees when it comes to making predictions

1. Binary Logistic Regression

- Used when the response is binary (i.e., it has two possible outcomes).
- Probability is modeled by logistic(sigmoid) function.
  - $\pi = \frac{\exp(X\beta)}{1+\exp(X\beta)} = \frac{1}{1+\exp(-X\beta)}$



2. Odds

- $\pi = \frac{\exp(X\beta)}{1+\exp(X\beta)} = \frac{1}{1+\exp(-X\beta)}$
- Odds: $\frac{\pi}{1-\pi} = \frac{\frac{\exp(X\beta)}{1+\exp(X\beta)}}{\frac{1}{1+\exp(X\beta)}} = \exp(X\beta)$
- Log Odds (logit transformation) : $\log\left(\frac{\pi}{1-\pi}\right) = X\beta$
- Logit function is the inverse of sigmoid function.

3. Logit Transformation

- Log Odds (logit transformation) : $\log\left(\frac{\pi}{1-\pi}\right) = X\beta$
  - Logit function is the inverse of sigmoid function.
  - Logit function maps probability to real numbers in $(-\infty, +\infty)$.

4. Likelihood

- $Pr(O|\theta)$
  - The joint probability of the observed data with the given parameters
- Likelihood for a binary logistic regression
  - $L(\beta; \mathbf{y}, X) = \prod_{i=1}^{n} \pi_i^{y_i}(1-\pi_i)^{1-y_i} =$

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

4. Likelihood

- Likelihood for a binary logistic regression
  - $L(\beta; \mathbf{y}, X) = \prod_{i=1}^{n} \pi_i^{y_i}(1-\pi_i)^{1-y_i}$
  - $0.6^1(1-0.6)^{1-1} \times 0.6^1(1-0.6)^{1-1} \times 0.6^0(1-0.6)^{1-0} = 0.6 \times 0.6 \times 0.4 = 0.144$
  - $0.3^1(1-0.3)^{1-1} \times 0.3^1(1-0.3)^{1-1} \times 0.3^0(1-0.3)^{1-0} = 0.3 \times 0.3 \times 0.7 = 0.063$

i	$\pi_i^{(1)}$	$\pi_i^{(2)}$	y_i
1	0.6	0.3	1
2	0.6	0.3	1
3	0.6	0.3	0

5. Log Likelihood

- Log likelihood for a binary logistic regression
  - $L(\beta; \mathbf{y}, X) = \prod_{i=1}^{n} \pi_i^{y_i}(1-\pi_i)^{1-y_i}$
  - $l(\beta) = \sum_{i=1}^{n}(y_i \log(\pi_i) + (1-y_i)\log(1-\pi_i)) = \sum_{i=1}^{n}(y_i X_i \beta - \log(1+\exp(X_i\beta)))$

5. Log Likelihood

- Coefficients are decided by maximizing the likelihood (or log likelihood) of the given training dataset.
    - Since log is a monotonic transformation, maximizing log likelihood results in maximizing likelihood.
    - Log likelihood, instead of likelihood, is used for optimization because it helps numerically.

5. Log Likelihood

- Maximizing likelihood (or log likelihood) has no closed-form solution.
  - Negative log likelihood (i.e. $-l(\beta)$) can be thought as a loss function.
  - IRLS (Iteratively Reweighted Least Squares) or quasi-Newton method can be used to estimate the parameters of a logistic regression model.

6. Data: Leukemia Remission

- Data: Leukemia Remission
  - y (REMISS): whether leukemia remission occurred
  - x_1 (CELL): cellularity of the marrow clot section
  - x_2 (SMEAR): smear differential percentage of blasts
  - x_3 (INFIL): percentage of absolute marrow leukemia cell infiltrate

6. Data: Leukemia Remission

- x_4 (LI): percentage labeling index of the bone marrow leukemia cells
- x_5 (BLAST): absolute number of blasts in the peripheral blood
- x_6 (TEMP): the highest temperature prior to start of treatment

7. Creating Logistic Regression Model

```
leukemia <- read.table("leukemia_remission.txt", header=T)
attach(leukemia)

model.2 <- glm(REMISS ~ LI + TEMP, family="binomial")
```

8. Prediction

```
> predict(model.2,
+         newdata=data.frame(LI=c(1.1, 0.7),
+                            TEMP=c(0.98, 0.99)),
+         type="link")
          1           2
 0.07873964 -1.71965085
> predict(model.2,
+         newdata=data.frame(LI=c(1.1, 0.7),
+                            TEMP=c(0.98, 0.99)),
+         type="response")
        1         2
0.5196747 0.1519161
```

8. Prediction

```
> prob = predict(model.2,
+          newdata=data.frame(LI=c(1.1, 0.7),
+                             TEMP=c(0.98, 0.99)),
+          type="response")
> pred = ifelse(prob > 0.5, 1, 0)
> prob
        1         2
0.5196747 0.1519161
> pred
1 2
1 0
```

9. Model Summary

```
> summary(model.2)
...
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   44.932     46.389   0.969   0.3327
LI             3.260      1.338   2.437   0.0148 *
TEMP         -49.428     47.386  -1.043   0.2969
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

10. Interpretation of Coefficients

- Odds: $\frac{\pi}{1-\pi} = \exp(X\beta)$
  - $\frac{\exp\left(\beta_0+\beta_1(x_1+1)\right)}{\exp(\beta_0+\beta_1 x_1)} = \exp(\beta_1)$
  - The odds increase multiplicatively by $\exp(\beta_\mathrm{k})$ for every one-unit increase in x_k.

11. Wald Test

- The Wald test is the test of significance for individual regression coefficients in logistic regression (recall that we use t-tests in linear regression).

  - $Z = \dfrac{\widehat{\beta}_i}{s.e.(\widehat{\beta}_i)}$

11. Wald Test

```
> summary(model.2)
...
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   44.932     46.389   0.969   0.3327
LI             3.260      1.338   2.437   0.0148 *
TEMP         -49.428     47.386  -1.043   0.2969
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
> 2 * pnorm((3.26/1.338), lower.tail=FALSE)
[1] 0.0148313
```

12. Model Summary

```
> summary(model.2)
...
    Null deviance: 34.372  on 26  degrees of freedom
Residual deviance: 24.817  on 24  degrees of freedom
AIC: 30.817
```

13. Deviance Test

- The likelihood ratio test is used to test the null hypothesis that any subset of the β's is equal to 0.
- This test procedure is analogous to the general linear F test procedure for multiple linear regression. However, note that when testing a single coefficient, the Wald test and likelihood ratio test will not, in general, give identical results.

13. Deviance Test

- $deviance = -2 \times (l(\hat{\beta}^{(0)}) - l(\hat{\beta}))$
  - $l(\hat{\beta}^{(0)})$: log likelihood of the (reduced) model specified by the null hypothesis
  - $l(\hat{\beta})$: log likelihood of the fitted (full) model

13. Deviance Test

- Wilk's theorem
  - As the sample size approaches ∞, the distribution of the test statistic $-2\log(\Lambda)$ asymptotically approaches the chi-square distribution.
    - Λ: likelihood ratio
    - degrees of freedom: difference in dimensionality of full parameter space and the subset of the parameter space

13. Deviance Test

```
> anova(model.2, test="Chisq")
Analysis of Deviance Table
Model: binomial, link: logit
Response: REMISS

Terms added sequentially (first to last)

     Df Deviance Resid. Df Resid. Dev Pr(>Chi)
NULL                    26     34.372
LI    1   8.2988        25     26.073 0.003967 **
TEMP  1   1.2564        24     24.817 0.262338
```

$-2 \times$ log-likelihood

13. Deviance Test

```
> pchisq(8.2988, df=1, lower.tail=FALSE)
[1] 0.003967128
> pchisq(1.2564, df=1, lower.tail=FALSE)
[1] 0.2623336
```

14. R Squared

- $R^2 = 1 - \frac{l(\hat{\beta})}{l(\hat{\beta}^{(0)})} = 1 - \frac{-2 \times l(\hat{\beta})}{-2 \times l(\hat{\beta}^{(0)})} = 1 - \frac{D(\hat{\beta})}{D(\hat{\beta}^{(0)})}$

14. R Squared

```
> model.0 <- glm(REMISS ~ 1, family="binomial")
> logLik(model.0)
'log Lik.' -17.18588 (df=1)
> logLik(model.2)
'log Lik.' -12.40829 (df=3)

> 1 - (logLik(model.2)[1]/logLik(model.0)[1]) #r2
[1] 0.277995

> 1 - (model.2$deviance/model.0$deviance)
[1] 0.277995
```

15. Hat Values

- The hat matrix serves a similar purpose as in the case of linear regression – to measure the influence of each observation on the overall fit of the model.
- As before, we should investigate any observations with $h_{ii} > 3 \times \frac{p}{n}$.

15. Hat Values

```
> hatvalues(model.2)
         1          2          3          4          5 
0.14020365 0.10195702 0.12205503 0.06938289 0.17929192 
         6          7          8          9         10 
0.10892656 0.06163518 0.37006668 0.07877282 0.05914946 
        11         12         13         14         15 
0.06938289 0.16826245 0.06155564 0.06646529 0.06420771 
        16         17         18         19         20 
0.37006668 0.07247703 0.07629323 0.07052136 0.06420771 
        21         22         23         24         25 
0.06155564 0.08898420 0.07629323 0.12974719 0.13599398 ...
```

16. Studentized Residual

```
> rstudent(model.2)
         1          2          3          4          5 
 0.6049008  0.9831232 -0.8597953 -0.5855823  0.9441080 
         6          7          8          9         10 
-0.6406039  1.5492545 -1.8477681 -0.5430310 -0.1272743 
        11         12         13         14         15 
-0.5855823 -1.4611590 -0.2261759 -0.6789944 -1.0290278 
        16         17         18         19         20 
 1.0434274 -0.2664334 -0.7308874 -0.5029834  1.3959439 
        21         22         23         24         25 
-0.2261759 -0.2464109  1.8072432 -1.8577716  0.6495935 ...
```

17. Cook's Distance

```
> cooks.distance(model.2)
           1            2            3            4
0.0115675816 0.0240100611 0.0214743673 0.0047835924
           5            6            7            8
0.0427249990 0.0096921975 0.0489804951 0.6321888885
           9           10           11           12
0.0046839816 0.0001755707 0.0047835924 0.1215294218
          13           14           15           16
0.0005839720 0.0063091341 0.0161301972 0.1528594454
          17           18           19           20
0.0009751784 0.0086611145 0.0035179838 0.0370317558 ...
```

Thank You