3장 로지스틱 회귀

로지스틱 회귀

- 로지스틱 회귀모형

  반응변수가 범주형(정성변수)인 경우에 적용되는 회귀분석 모형

  예측모형: 설명변수의 값이 주어질 때 반응변수의 각 집단에 속할 확률이 얼마인지를 추정

  분류모형: 추정확률의 기준치에 따라서 분류의 목적으로 사용 가능

- 종류

  이진형: 이항 로지스틱 회귀 모형

  다범주형: 다범주 로짓 회귀 모형

  순서형: 순서형 로지스틱 회귀 모형

로지스틱 회귀

- 단순 로지스틱 회귀모형 배경(베르누이 분포의 회귀분석 접근)

[베르누이 분포]

$$Y \sim Bernoulli(\pi)$$

$$\pi = P(success) = P(Y = 1)$$

[회귀분석 접근법]

$$\pi(x) = P(Y = 1|X = x)$$

$$\Rightarrow E(Y|x) = \sum_{all\ y} y \times P(y|x) = 1 \times P(Y = 1|x) + 0 \times P(Y = 0|x)$$

$$\neq \alpha + \beta x$$

Note: $0 \leq \pi(x) \leq 1$, $-\infty < \alpha + \beta x < \infty$

로지스틱 회귀

- 단순 로지스틱 회귀모형 형태

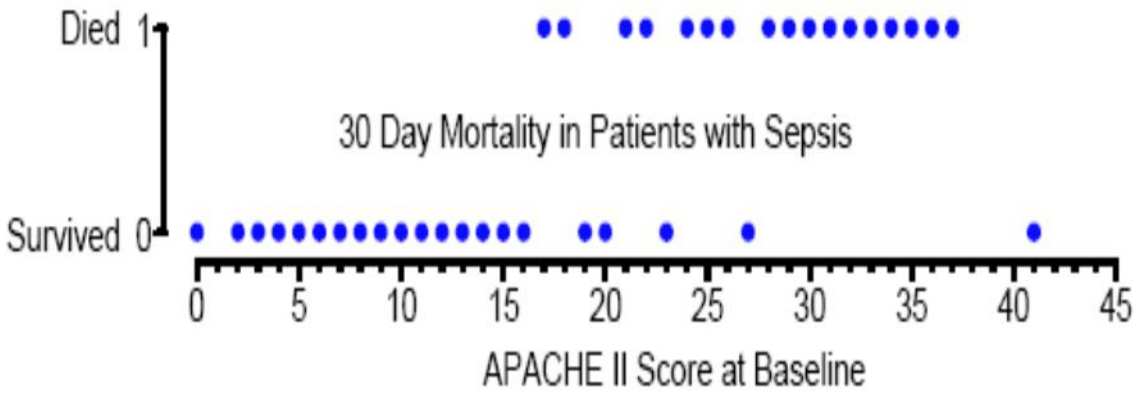
$$\text{logit}(\pi(x)) = \ln\left(\frac{\pi(x)}{1-\pi(x)}\right) = \alpha + \beta x$$

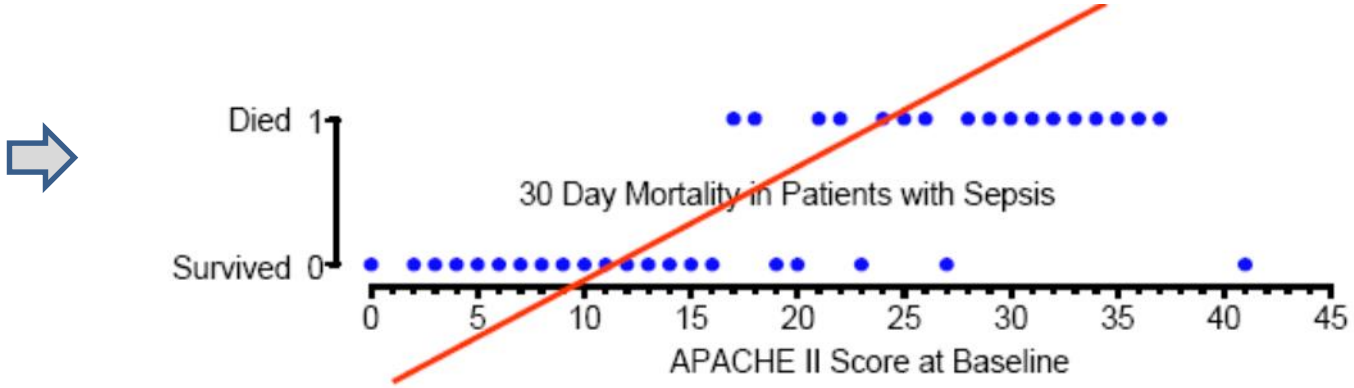
여기서 $\pi(x) = P(Y = 1|x)$ 이고, Y 는 0 또는 1을 갖는 반응 변수이고, x 는 설명변수이다.

$$\pi(x) = \frac{\exp(\alpha + \beta x)}{1 + \exp(\alpha + \beta x)} = \frac{1}{1 + \exp(-[\alpha + \beta x])}$$

로지스틱 회귀

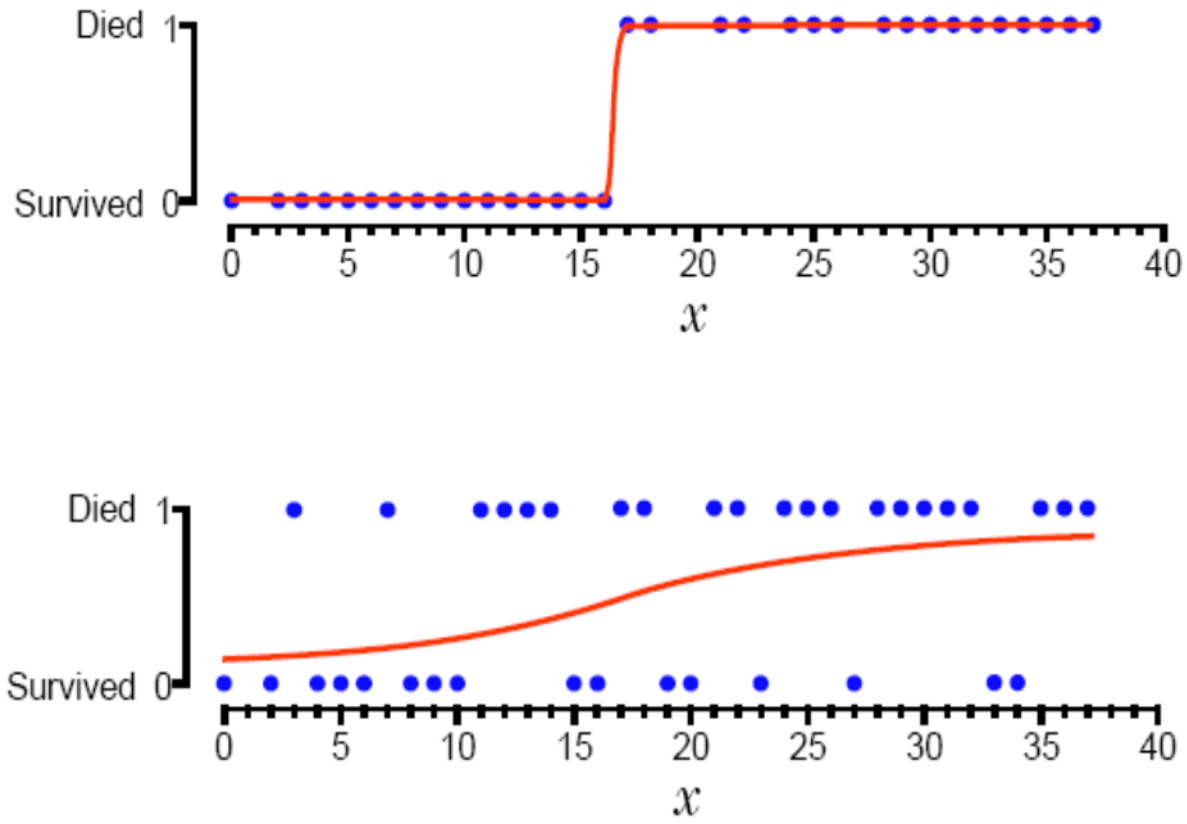
- 단순 회귀 모형 적용





로지스틱 회귀

- 로지스틱 회귀 모형 적용: $\beta > 0$



로지스틱 회귀

- 상대위험(relative risk): 비율

예) π_1 을 중소기업의 부도확률, π_2 을 대기업의 부도확률이라고 하자.

또는 $\pi_1 = P(Y = 1|X \in \text{중소기업}(1))$, $\pi_2 = P(Y = 1|X \in \text{대기업}(2))$ 라고 하자.

비율이 0과 1 근처일 때 훨씬 중요한 의미를 갖는다.

π_1=0.46, π_2=0.452 인 경우와 π_1=0.01, π_2=0.002에서 두 비율의 차이는 0.008로 같지만, 상대위험은 후자가 훨씬 크다.

- Odds와 Odds Ratio

상대위험의 문제점을 극복하기 위해서 개발

$$odds_1 = \frac{\pi_1}{1-\pi_1} \quad , \quad odds_2 = \frac{\pi_2}{1-\pi_2}$$

Odds ratio for 1 to 2: = $\frac{odds_1}{odds_2} = \frac{\pi_1/(1-\pi_1)}{\pi_2/(1-\pi_2)}$

❖ $odds_1 = odds_2$ 이면, odds ratio=1이고, π_1=π_2가 성립됨

로지스틱 회귀

- 로지스틱 회귀 계수(coefficient) 해석

$$\alpha = \ln\left(\frac{\pi(0)}{1-\pi(0)}\right)$$: X=0일 때 log of odds

$$\beta = \ln\left(\frac{\pi(1)/(1-\pi(1))}{\pi(0)/(1-\pi(0))}\right)$$: log of odds ratio for X=1 to X=0

Note) $$\ln\left(\frac{\pi(x)}{1-\pi(x)}\right) = \alpha + \beta x$$

$$\ln\left(\frac{\pi(1)}{1-\pi(1)}\right) = \alpha + \beta \quad , \quad \ln\left(\frac{\pi(0)}{1-\pi(0)}\right) = \alpha$$

로지스틱 회귀

- 다중 로지스틱 회귀모형 형태

$$\ln\left(\frac{\pi(x)}{1-\pi(x)}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k$$

여기서 $\pi(x) = P(Y = 1|x)$ 이고, $x = (x_1, \ldots, x_k)$ 이다.

$$\pi(x) = \frac{\exp(\beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k)}{1 + \exp(\beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k)}$$
$$= \frac{1}{1 + \exp(-[\beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k])}$$

로지스틱 회귀

- 프로빗 (Probit) 모형 적용

로지스틱 회귀 모형에서 $\pi(x)$ 은 로지스틱 분포의 누적분포함수(CDF)에 해당됨

$$\pi(x) = F(\beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k)$$

프로빗 모형은 위의 $F(\cdot)$ 대신 표준정규분포의 누적분포함수 $\Phi(\cdot)$ 로 성공의 확률을 모형화한 것임. 즉, 아래와 같음

$$\Phi^{-1}(\pi(x)) = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k$$

로지스틱 회귀

- 분류 목적

  로지스틱 회귀가 분류의 목적으로 사용될 경우에는 일반적으로 $\pi(x)$=P(Y=1|x)가 기준값 0.5 보다 크면, Y=1인 집단으로 분류하고, 0.5 보다 작으면, Y=0인 집단으로 분류

- 베이즈 정리 이용

$$\pi(x_0) = P(\theta_1 \mid x_0) = \frac{P(\theta_1 \ occurs \ and \ we \ observe \ x_0)}{P(we \ observe \ x_0)}$$

$$= \frac{P(we \ observe \ x_0 \mid \theta_1)P(\theta_1)}{P(we \ observe \ x_0 \mid \theta_1)P(\theta_1) + P(we \ observe \ x_0 \mid \theta_2)P(\theta_2)} = \frac{p_1 f_1(x_0)}{p_1 f_1(x_0) + p_2 f_2(x_0)}$$

여기서 $P(\theta_1 \mid x_0) + P(\theta_2 \mid x_0) = 1$ 성립

Rule: 만약 $P(\theta_1 \mid x_0) > P(\theta_2 \mid x_0)$ 이면, new data를 그룹 1로 분류

⇔ 만약 $P(\theta_1 \mid x_0) > 0.5$ 이면, new data를 그룹 1로 분류

⇔ 만약 $p_1 f_1(x_0) > p_2 f_2(x_0)$ 이면, new data를 그룹 1로 분류

로지스틱 회귀

- Expected Cost of Misclassification (ECM)

$$ECM = c(2|1)P(2|1)p_1 + c(1|2)P(1|2)p_2$$

➡ Rule to minimize ECM:

만약 $\frac{f_1(x)}{f_2(x)} \geq \frac{c(1|2)}{c(2|1)} \times \frac{p_2}{p_1}$ 이면, new data를 그룹 1로 분류

➡ 특수 가정 ($c(2|1) = c(1|2)$) 하의 Rule:

만약 $\frac{f_1(x)}{f_2(x)} \geq \frac{p_2}{p_1}$ 이면, new data를 그룹 1로 분류

p_1: prior probability of belonging to group 1

p_2: prior probability of belonging to group 2

$P(2|1)$: probability of incorrectly classifying group 1 as group 2

$P(1|2)$: probability of incorrectly classifying group 2 as group 1

$c(2|1)$: cost of incorrectly classifying group 1 as group 2

$c(1|2)$: cost of incorrectly classifying group 2 as group 1

로지스틱 회귀

- 예제1: iris 자료의 이항 로지스틱 모형 분류

```
data(iris);
a=subset(iris, Species=="setosa"| Species =="versicolor"); # species가 setosa와 versicolor인 자료만 추출
a$Species=factor(a$Species) ;
str(a) ;                          # setosa를 1 (Y=0), versicolor을 2 (Y=1)로 처리
head(a);

b=glm(Species~Sepal.Length, data=a, family=binomial);  # 로지스틱 회귀모형 피팅
summary(b);
#coef(b);                          # sepal.lengt가 1단위 증가함에 따라 versicolor일 오즈가 exp(5.14)배 증가
#exp(coef(b)["Sepal.Length"]);

#confint(b, parm="Sepal.Length");        # 회귀계수 b1 (log 오즈비)의 신뢰구간 구하기: 0 포함 여부 확인
#exp(confint(b, parm="Sepal.Length"));  # 회귀계수 exp(b1) (오즈비)의 신뢰구간 구하기: 1 포함 여부 확인
fitted(b);                                  # π(x):  fitted(b)는 predict(b, type="response")와 같음.
fitted(b)[c(1:5, 96:100)];                  # 적합결과

predict(b, newdata=a[c(1,50,51,100),],type="response");
    # (확률) 예측결과: 0.5 보다 크면, versicolor로 분류 (새로운 데이터 셋을 활용할 수 있음)
cdplot(Species~Sepal.Length, data=a);      # sepal.length가 커짐에 따라 versicolor 확률이 커짐.
plot(a$Sepal.Length, a$Species, xlab="Sepal.Length");
x=seq(min(a$Sepal.Length), max(a$Sepal.Length), 0.1);
lines(x, 1+(1/(1+(1/exp(-27.831+5.140*x)))), type="l", col="red");    # 로지스틱 회귀모형 그래프
```

로지스틱 회귀

- 예제2: 1973~1974년도에 생산된 32종류의 자동차에 대한 분류

```
attach(mtcars);
head(mtcars);
str(mtcars) ;            # vs가 이항변수 (flat 엔진=0, straight 엔진=1)
glm.vs=glm(vs~mpg+am+gear, data=mtcars, family=binomial);  # am (transmission=0, auto=1)
summary(glm.vs);

step.vs=step(glm.vs, direction="backward");  # backward 기법을 이용한 변수 선택
   # both, forward 이용 가능
summary(step.vs);

#ls(glm.vs);       # 제공 가능한 명령문 나열 예) glm.vs$fitted.values
#str(glm.vs);

#anova(glm.vs, test="Chisq");
# 두 변수(mpg와 am)를 차례로 포함 시 이탈도 감소량이 유의적인지 보여줌
#1-pchisq(18.327,1);        # p-value 계산
#1-pchisq(4.887,1);         # p-value 계산
```

로지스틱 회귀

- 예제3: German credit dataset 이용

```
credit.df=read.csv("german_credit_dataset.csv", header=TRUE, sep=",");

# 데이터형 변환 함수(factor와 정규화)
to.factors=function(df, variables) {
                for(variable in variables)  {
                        df[[variable]]=as.factor(df[[variable]])
}
return(df)
};
scale.features=function(df, variables) {
                 for(variable in variables)  {
                        df[[variable]]=scale(df[[variable]], center=T, scale=T)
}
return(df)
} ;

# 데이터 변환할 때 변수선택
categorical.vars=c('credit.rating','account.balance','previous.credit.payment.status', 'credit.purpose',
'savings','employment.duration','installment.rate','marital.status','guarantor', 'residence.duration',
'current.assets','other.credits','apartment.type','bank.credits','occupation', 'dependents', 'telephone',
'foreign.worker');

numeric.vars=c("credit.duration.months","age","credit.amount");
```

로지스틱 회귀

- 예제3: German credit dataset 이용

```
# 데이터형 변환
credit.df1=to.factors(df=credit.df, variables=categorical.vars);  # 범주형 변수 factor화
credit.df2=scale.features(credit.df1, numeric.vars);                # 정규화


# 데이터를 train: test로 60:40 비율로 분리
set.seed(1234);
indexes=sample(1:nrow(credit.df2), size=0.6*nrow(credit.df2));    # default: 비복원추출
train.data=credit.df2[indexes, ];
test.data=credit.df2[-indexes, ];

library(caret);
library(ROCR);
# source("performance_plot_utils.R"); # plotting metric results
# 특징과 클라스 변수를 분리
test.feature.vars=test.data[ ,-1];    # test 데이터의 1번째 변수를 제거 (독립변수만 남김)
test.class.var=test.data[ ,1];    # test 데이터의 1번째 변수만으로 구성 (종속변수: credit rating)

# 초기 모델 훈련
formula.init="credit.rating~." ;# ~. 모든 변수 포함
formula.init=as.formula(formula.init);
lr.model=glm(formula=formula.init, data=train.data, family="binomial"); summary(lr.model);
                                                    # train 데이터로 fitting
```

로지스틱 회귀

- 예제3: 계속

```
# test data로 예측 후 결과 평가(기준치를 0.5로 적용)
lr.predictions=predict(lr.model, test.data, type="response");
lr.predictions1=round(lr.predictions);       # 0.5 이하면 0으로 분류, 0.5 초과면 1로 분류
confusionMatrix(data=lr.predictions1, reference=test.class.var, positive="1");
    # reference가 실제 값. lr.predictions1가 예측 값.

# test data로 예측 후 결과 평가(기준치를 변경하면서 적용)
lr.model.best = lr.model;
lr.prediction.values = predict(lr.model.best, test.feature.vars, type="response");
   # test data의 독립변수 이용하여 y 예측
pred = prediction(lr.prediction.values, test.class.var);
  # test data의 실제 y와 예측으로 구한 y를 비교한 정보 종합: ROC 커브 작성 위한 자료
pred ;

Roc=performance(pred, "tpr", "fpr");           # Roc 커브: true positive(y)/ false positive(x)
Recall=performance(pred, "prec", "rec");       # Precision/ Recall 커브: prec(y)/ rec(x)
SenSpec=performance(pred, "sens", "spec"); # Sensitivity/ Specification 커브: sens(y)/ spec(x)
area_under_curve=performance(pred, "auc"); # AUC

x11();
par(mfrow=c(2,2));
plot(Roc, title.text="LR ROC Curve");
plot(Recall, title.text="LR Precision/Recall Curve");
plot(SenSpec, title.text="LR Sensitivity/Specification Curve");
```

Model Evaluation for classification

❖ **Model evaluation**

Binary data case

Confusion matrix		Predicted group	
		Y=1	Y=0
Real group	Y=1	f11(true positive)	f12(false negative)
	Y=0	f21(false positive)	f22(true negative)

① Accuracy or Correct classification rate: Corrected predicted ratio out of the whole (f11+f22)/n

② Sensitivity: Ratio of predicting true when they are true f11/(f11+f12)

③ Specificity: Ratio of predicting false when they are false f22/(f21+f22)

- ❖ Accuracy = (f11+f12)/n X Sensitivity + (f21+f22)/n X Specificity
- ❖ Error rate = 1 – Accuracy

Model Evaluation for classification

❖ **Consideration**

(1) Cross validation

Split the whole data into train data set and test data set

Train data set is for model building, and test data set is for model evaluation

If the whole data are large enough, split the train: test into 50:50 randomly

a) K-fold

① If the data are not large enough, split the whole data into k groups.

k-1 groups are used for model building, 1 group is used for model validation.

② This procedure is repeated k times, and calculate the average accuracy.

b) Leave one out method

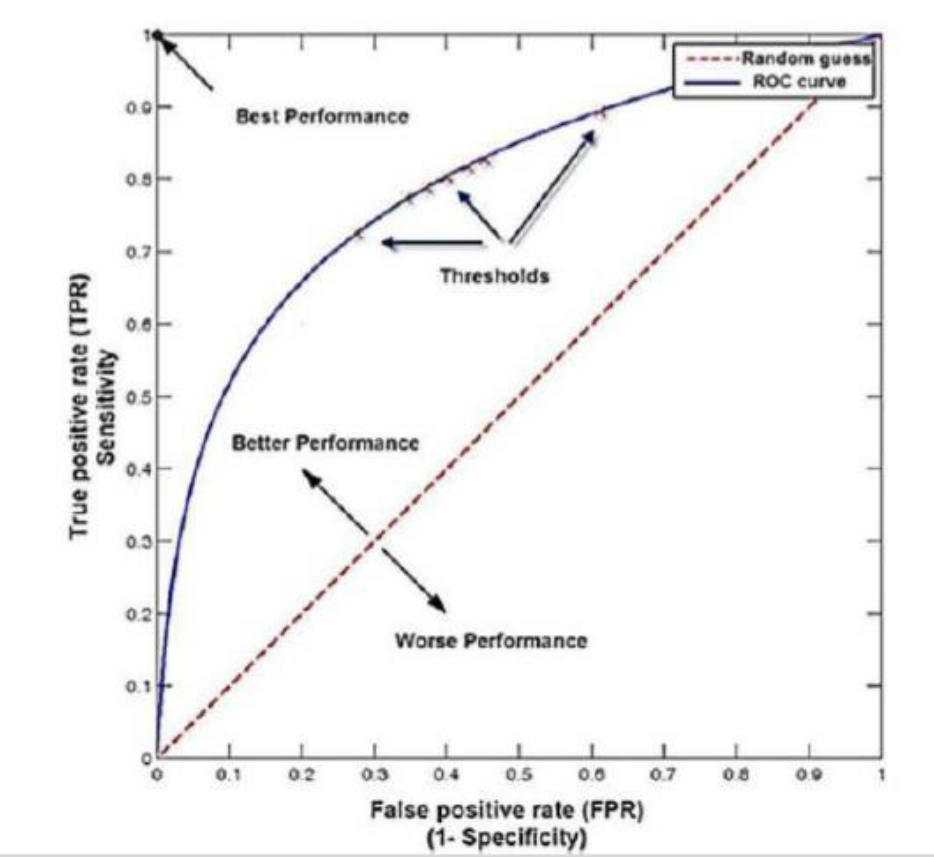
① Groups k = n (sample size) is treated. I.e., only one data point is used for model evaluation and all the others are for model building.

② This procedure is repeated n times, and calculate the average accuracy.

Model Evaluation for classification

(2) ROC (Receiver Operating Characteristic) curve (useful for binary classification)

- The threshold is changed from 0 to 1, and build the confusion matrix for each threshold.
- For each threshold, you can compute a pair of {sensitivity, specificity}. Using the pairs, draw the ROC curve.



AUC score	
0.9~1.0	High value
0.8~0.9	
0.7~0.8	
0.6~0.7	
0.5~0.6	Low value
Below 0.5	Valueless