Machine Learning in Public Health

Lecture 3: Logistic Regression, LDA, QDA

Dr. Yang Feng

Today's agenda

- Logistic Regression
- Linear Discriminant Analysis

Classification

- **Classification**: supervised learning when outcomes are categorical (a.k.a. qualitative)
- The categorical outcomes (responses) are usually called *class labels*.
- Both *classification* and *regression* are supervised learning
- Classification is perhaps the *most* widely used machine learning methods.

Classification (Examples)

1. A person arrives at the emergency room with a set of symptoms that could possibly be attributed to one of three medical conditions. Which of the three conditions does the individual have?

2. An online banking service must be able to determine whether or not a transaction being performed on the site is fraudulent, on the basis of the user's IP address, past transaction history, and so forth.

3. On the basis of DNA sequence data for a number of patients with and without a given disease, a biologist would like to figure out which DNA mutations are deleterious (disease-causing) and which are not.

Classification

- There are many off-the-shelf classification methods. In this lecture, we will cover two most common (basic) ones:
  - *logistic regression*
  - *linear discriminant analysis (LDA)*

What is the usual objective for classification?

- Binary classification is the most common classification scenario
- Features $X \in R^p$ and class labels $Y \in \{0, 1\}$
- A classifier h is some function (usually data-dependent function) that maps the feature space into the label space. One can think of a classifier as a data-dependent partition of the feature space
- The classification error (risk) is the probability of misclassification. In other words:

  $P(h(X) \neq Y)$, where P is regarding the joint distribution of (X, Y) .

- $P(h(X) \neq Y)$ is usually denoted by $R(h)$
- *Often* (NOT always), we construct classifiers to minimize the classification error.

Why not linear regression?

- A general remark: before inventing new methods, we should ask why the existing ones do not suffice

- Suppose that we are trying to predict the medical condition of a patient in the emergency room on the basis of her symptoms.

- In this simplified example, there are three possible diagnoses: stroke, drug overdose, and epileptic seizure.

- We could consider encoding these values as a quantitative response variable, Y, as

  - $Y = 1$ *(if stroke)*
  - $Y = 2$ *(if drug overdose)*
  - $Y = 3$ *(if epileptic seizure).*

- Issues?

Why not linear regression?

- we have endorsed an ordering in the types
- we assumed the same difference between pairs
- an equally reasonable coding
  - $Y = 1$ *(if epileptic seizure)*
  - $Y = 2$ *(if stroke)*
  - $Y = 3$ *(if drug overdose)*
- will imply a totally different relationship among the three types and will lead to different predictions

Why not linear regression?

- When the outcome variable has 2 categories
  - *we can introduce* ***dummy variables***
  - *and cut the predicted Y's at some level, i.e., declare prediction above that level of class 1, and 0 otherwise*
- Issue: could predict negative probability!

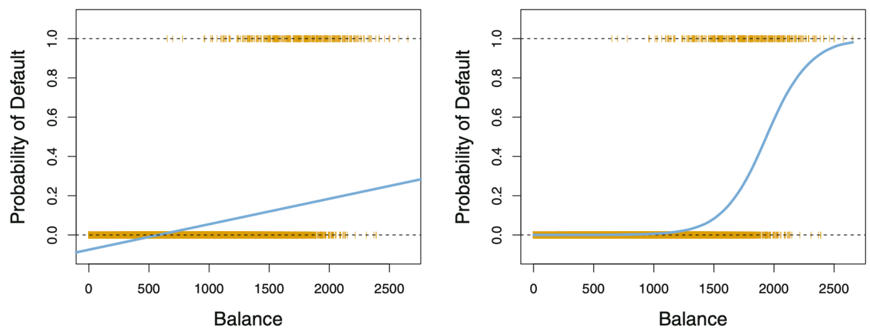


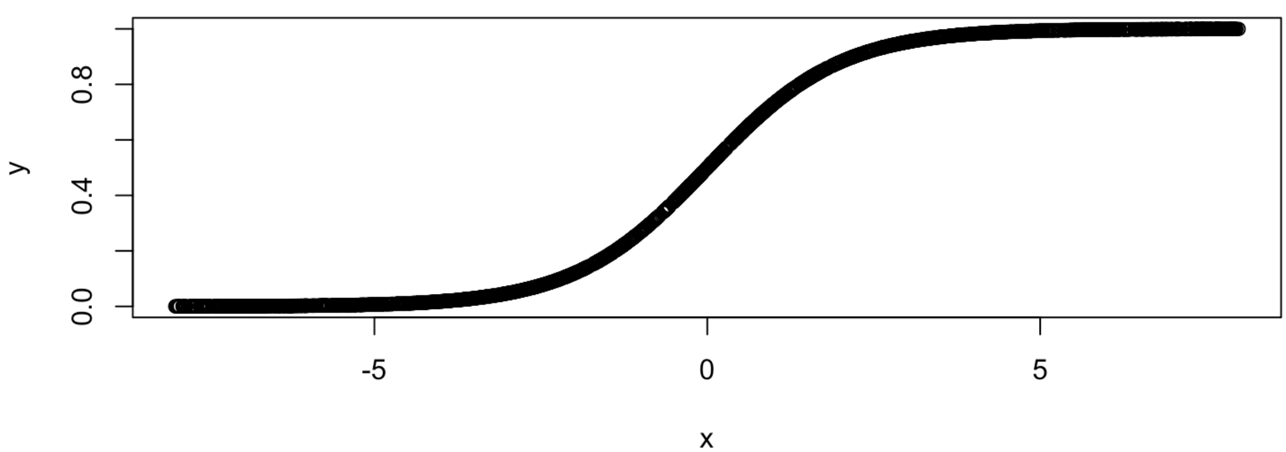
FIGURE 4.2. *Classification using the* `Default` *data.* Left: *Estimated probability of* `default` *using linear regression. Some estimated probabilities are negative! The orange ticks indicate the 0/1 values coded for* `default`*(*`No` *or* `Yes`*).* Right: *Predicted probabilities of* `default` *using logistic regression. All probabilities lie between* 0 *and* 1.

Logistic regression

- Model the conditional probability $P(Y = 1|X)$ (compare with linear model)
- The logistic (a.k.a. sigmoid) function

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

- Logistic regression model: $P(Y = 1|X) = f(X)$.
- Plot the sigmoid function when $\beta_0 = 0$ and $\beta_1 = 1$



Logistic regression

- The sigmoid function f takes values between 0 and 1; perfect for modeling probability
- Under the logistic regression model, the log-odds or logit is linear in the input variable X:

$$\log\left(\frac{P(Y=1|X)}{P(Y=0|X)}\right) = \beta_0 + \beta_1 X$$

- In some books, the above equation is the definition of logistic regression model, or called logit model. These two definitions are equivalent
- For logit function: https://en.wikipedia.org/wiki/Logit
- β_1 can be interpreted as the average change in log-odds associated with a one-unit increase in X
- β_1 does NOT correspond to the change in $P(Y=1|X)$ associated with a one-unit increase in X
- Q: recall the interpretation of the coefficients in linear regression, and find out the difference

Fitting Logistic regression

- The coefficients β_0 and β_1 in the sigmoid function are unknown
- Need to estimate them from training data
- Given training data (pairs are independent of each other)

$$\{(x_1, y_1), \cdots, (x_n, y_n)\}$$

- And let $p(x) = P(Y = 1|x)$. We would like to find $\hat{\beta}_0$ and $\hat{\beta}_1$ such that they maximize the likelihood function $\ell(\beta_0, \beta_1)$:

$$\ell(\beta_0, \beta_1) = \Pi_{i:y_i=1} p(x_i) \Pi_{j:y_j=0} [1 - p(x_j)]$$

- This is called a maximum likelihood approach
- The least squares method for linear regression is in fact also a maximum likelihood approach

About likelihood function

- Likelihood function is a frequentist idea

- To motivate the idea, suppose you know the distribution is $\mathcal{N}(\mu, 1)$ with some known parameter μ, and you have observed a few points in the neighborhood of 3, which μ has the best opportunity to have produced these points?

- Q: Given observations (x, y pairs): $\{(2, 1), (3, 0)\}$, write down the **likelihood function** $\ell(\beta_0, \beta_1)$ based on logistic regression model (Hint: start with only one pair $(2, 1)$)

- A: For observation $(2, 1)$.

$$\ell(\beta_0, \beta_1) = \frac{e^{\beta_0 + 2\beta_1}}{1 + e^{\beta_0 + 2\beta_1}}$$

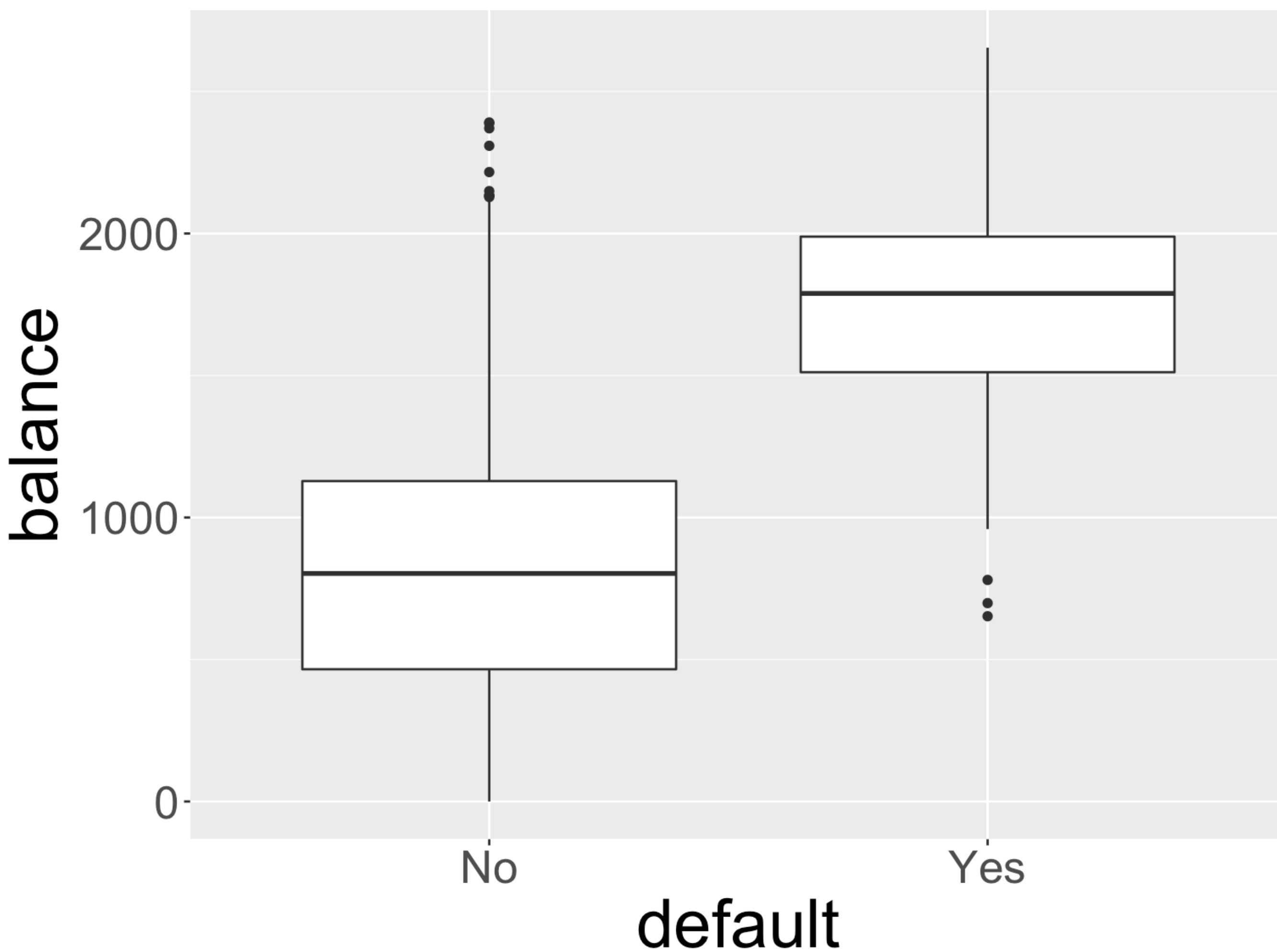
For $\{(2, 1), (3, 0)\}$, we have

$$\ell(\beta_0, \beta_1) = \frac{e^{\beta_0 + 2\beta_1}}{1 + e^{\beta_0 + 2\beta_1}} \frac{1}{1 + e^{\beta_0 + 3\beta_1}}$$

The `Default` dataset

	default	student	balance	income
	<fct>	<fct>	<dbl>	<dbl>
1	No	No	729.5265	44361.63
2	No	Yes	817.1804	12106.13
242	Yes	Yes	1572.8565	14930.18
243	No	No	0.0000	40150.22
244	Yes	No	1964.4769	39054.59

5 rows



From the boxplots, we guess that `balance` might have good prediction power

Use balance to predict default

- Split the data into training and testing

```
default_tr <- Default[1:3000, ]
default_te <- Default[-(1:3000), ]
```

- Fit the logistic regression model on the training data.

```
fit_balance <- glm(default ~ balance, data = default_tr, family='binomial');
summary(fit_balance)
```

```
##
## Call:
## glm(formula = default ~ balance, family = "binomial", data
= default_tr)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2785  -0.1349  -0.0511  -0.0170   3.6540
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.137e+01  7.190e-01  -15.81   <2e-16 ***
## balance      6.013e-03  4.405e-04   13.65   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 883.59  on 2999  degrees of freedom
## Residual deviance: 456.51  on 2998  degrees of freedom
## AIC: 460.51
##
## Number of Fisher Scoring iterations: 8
```

Use `balance` to predict `default`

- How good is the prediction on training data?

```
pred_train_prob <- predict(fit_balance, type = 'response')
pred_train_label <- ifelse(pred_train_prob > 0.5, 'Yes', 'No')
table(pred_train_label, default_tr$default) #Confusion Table
```

```
##
## pred_train_label   No  Yes
##              No  2884   66
##              Yes   15   35
```

- Now, we can get the training error.

```
mean(pred_train_label != default_tr$default)
```

```
## [1] 0.027
```

- The overall training classification error looks good.
- $1(P(Y = 1|X = x) \geq 1/2)$ is the Bayes classifier (a.k.a. oracle classifier) in binary classification

Use `balance` to predict `default`

- How good is the prediction on test data?

```
pred_test_prob <- predict(fit_balance, newdata = default_te, type = 'response')
pred_test_label <- ifelse(pred_test_prob > 0.5, 'Yes', 'No')
table(pred_test_label, default_te$default)
```

```
##
## pred_test_label   No  Yes
##             No  6724  145
##             Yes   44   87
```

- Now, we can get the test classification error.

```
mean(pred_test_label != default_te$default)
```

```
## [1] 0.027
```

- The overall test classification error looks also very good!

(Multiple) logistic regression

- Like linear regression, logistic regression can take vector inputs

- When $X \in R^p$, we model the log-odds as

$$\log\left(\frac{P(Y=1|X)}{P(Y=0|X)}\right) = \beta_0 + \beta_1 X_1 \cdots + \beta_p X_p$$

- Predict `default` using both `balance` and `student`:

```
fit_bal_stu <- glm(default ~ balance + student, data= default_tr,
        family='binomial')
```

(Multiple) logistic regression

```r
summary(fit_bal_stu)
```

```
##
## Call:
## glm(formula = default ~ balance + student, family = "binomial",
##     data = default_tr)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1660  -0.1284  -0.0463  -0.0149   3.6344
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.156e+01  7.432e-01  -15.55  < 2e-16 ***
## balance      6.347e-03  4.699e-04   13.51  < 2e-16 ***
## studentYes  -8.552e-01  2.777e-01   -3.08  0.00207 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 883.59  on 2999  degrees of freedom
## Residual deviance: 446.38  on 2997  degrees of freedom
## AIC: 452.38
##
## Number of Fisher Scoring iterations: 8
```

(Multiple) logistic regression: compute the training error

```
pred_train_prob <- predict(fit_bal_stu, type = 'response')
pred_train_label <- ifelse(pred_train_prob > 0.5, 'Yes', 'No')
table(pred_train_label, default_tr$default)
```

```
##
## pred_train_label   No  Yes
##              No  2887   65
##              Yes   12   36
```

```
mean(pred_train_label != default_tr$default)
```

```
## [1] 0.02566667
```

- This training classification error is smaller than that of the model using just `balance`
- But this is error on training data; a more fair comparison should be on new data (test data) that were not used to train the model
- Q: As model complexity increases, how should training error change? How about test error?

(Multiple) logistic regression: compute the test error

```
pred_test_prob <- predict(fit_bal_stu, newdata = default_te, type = 'response')
pred_test_label <- ifelse(pred_test_prob > 0.5, 'Yes', 'No')
mean(pred_test_label != default_te$default)
```

```
## [1] 0.027
```

- The same error as only using the `balance`.

Extensions to multi-label logistic regression

- one versus all approach
- multinomial regression

Exercise

Suppose we collect data for a group of students in a biostatistics class with variables

- X_1: hours studied,
- X_2: undergrad GPA,
- Y: receive an A.
- We fit a logistic regression and produce estimated coefficients $\hat{\beta}_0 = -6, \hat{\beta}_1 = 0.05, \hat{\beta}_2 = 1$.
- Question:
  - *Estimate the probability that a student who studies for 40 hours and has an undergrad GPA of 3.5 to get an A in the class.*
  - *How many hours would the student in the previous part need to study to have a 50% chance of getting an A in the class?*

Two different modeling approaches

- Logistic regression models $P(Y = 1|X)$ by the logistic function
- Linear discriminant analysis (LDA) models the conditional distribution of X given Y by Gaussian (a.k.a. normal) distributions
- Bayes' Theorem
  - *Suppose π_k denotes prior probability $P(Y = k)$ of the kth class*
  - *Suppose $f_k(X)$ denotes the density function of X for an observation that comes from the kth class (i.e. f_k is the p.d.f. of $(X|Y = k)$)*
  - *Then we have*

$$P(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)},$$

where K is the total number of classes

Bayes Classifier

- If a classifier assigns an observation x to class k that has the largest $p_k(x) = P(Y = k|X = x)$, this classifier has the minimum classification error (Bayes classifier, a.k.a. oracle classifier).
- In other words, Bayes classifier assign x to $\arg\max \pi_k f_k(x)$
- Often, estimating π_k's is straightforward; but estimating f_k involves some technicality

Linear discriminant analysis (LDA) (for $p = 1$)

- In LDA, we assume that $f_k(x)$'s are normally distributed, and they have the same standard deviation coefficient:

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x - \mu_k)^2 \right)$$

Linear discriminant analysis (LDA) (for $p = 1$)

- Under this assumption, it can be shown that assigning an observation to a class according to the largest $p_k(x)$ is the same as assigning to the observation according to the largest

$$\delta_k(x) = x \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k)$$

- Note that the δ_k function is linear in x. The LDA approximates $\delta_k(x)$ by plugging estimates for π_K, μ_k and σ^2:

$$\hat{\pi}_k = n_k/n, \; \hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i$$

where n_k is the number of training observations in the k-th class

$$\hat{\sigma}^2 = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)^2$$

Linear discriminant analysis (LDA)

- So for $p = 1$ the **discriminant function** is

$$\hat{\delta}_k(x) = x \cdot \frac{\hat{\mu}_k}{\hat{\sigma}^2} - \frac{\hat{\mu}_k^2}{2\hat{\sigma}^2} + \log(\hat{\pi}_k)$$

  $\hat{\delta}_k(x)$ is linear in x

- LDA classifier: assign x to class k for the largest $\hat{\delta}_k(x)$
- Q: for $K = 2$, show LDA has a linear decision boundary.

Linear discriminant analysis (LDA, $p > 1$)

- When $p > 1$, assume $f_k(X) \sim \mathcal{N}(\mu_k, \Sigma)$, a multivariate normal distribution with mean μ_k and covariance matrix Σ (Equation (4.23) in `ISLR` is the probability density function of a multivariate normal variable)
- Bayes classifier assigns an observation to class k for the largest

$$\delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log \pi_k$$

Again, need to plug-in estimates of μ_k, Σ and π_k

Analyzing Default Data using LDA

```r
library(MASS) # lda() is in `MASS` library
lda.fit <- lda(default ~ balance, data = default_tr)
lda.fit
```

```
## Call:
## lda(default ~ balance, data = default_tr)
##
## Prior probabilities of groups:
##         No        Yes
## 0.96633333 0.03366667
##
## Group means:
##       balance
## No   798.3913
## Yes 1750.5748
##
## Coefficients of linear discriminants:
##                 LD1
## balance 0.002228241
```

Analyzing Default Data using LDA

- Training Error

```
lda.predict <- predict(lda.fit, default_tr)
lda.class <- lda.predict$class
mean(lda.class != default_tr$default)
```

```
## [1] 0.02933333
```

- Test Error

```
lda.predict <- predict(lda.fit, default_te)
lda.class <- lda.predict$class
mean(lda.class != default_te$default)
```

```
## [1] 0.02685714
```

Quadratic discriminant analysis (QDA)

- Model

$$(X|Y = k) \sim \mathcal{N}(\mu_k, \Sigma_k)$$

- Compared to LDA, QDA allows (potentially) different covariance matrices

- The decision boundary of QDA is (most likely) not linear

- A quadratic function in x appears in the discriminant function

- In practice, QDA is not as popular as LDA

Analyzing Default Data using QDA

```r
qda.fit <- qda(default ~ balance, data = default_tr)
qda.fit
```

```
## Call:
## qda(default ~ balance, data = default_tr)
##
## Prior probabilities of groups:
##         No        Yes
## 0.96633333 0.03366667
##
## Group means:
##       balance
## No   798.3913
## Yes 1750.5748
```

Analyzing Default Data using QDA

- Training Error

```
qda.predict <- predict(qda.fit, default_tr)
qda.class <- qda.predict$class
mean(qda.class != default_tr$default)
```

```
## [1] 0.02833333
```

- Test Error

```
qda.predict <- predict(qda.fit, default_te)
qda.class <- qda.predict$class
mean(qda.class != default_te$default)
```

```
## [1] 0.02714286
```

Next Class

We will continue discussion about classification

- K-Nearest Neighbors (KNN)
- Receiver operating characteristics (ROC) curves
- Imbalanced classes and asymmetric errors