Supervised Learning: Classification

Logistic Regression and Discriminant Analysis

Maarten Cruyff

Program

Morning session

Logistic regression and LDA

Classification refers to supervised learning for categorical outcome variables. The aim of classification is to predict the class an observations belongs to based on a set of features. This session introduces two classical methods for classification; logistic regression and linear discriminant analysis. The logistic regression model applies to outcomes with two classes. It estimates the probability of a "success" on the outcome variable, and classifies an observation as a success when this probability exceeds a certain threshold. Linear discriminant analysis is also suited for outcome variables with more than two classes, and it estimates linear discriminant functions that optimally separate between the classes. Since the MSE is not a useful fit criterion for classifying observations, model performance is evaluated with the confusion matrix, the AIC and/or a ROC curve,

Course materials

- Lecture sheets
- R lab
- R Markdown lab template

Categorical outcomes

How to make the following predictions:

- What will be a person's voting behavior given a set of background variables?
- How to diagnose a patient given a set of symptoms?
- What are the predictors for successfully stopping with smoking?

These questions involve the prediction of a class and not of a score.

The linear model is unsuited for this purpose, but what is?

Content

1. Logistic regression
2. Discriminant analysis
3. Classification criteria

What's classification?

Outcome variable is categorical

- Predict **class membership** from feature set

Estimation

1. Estimate $P(class = j \mid features)$
2. Assign observation to class with largest probability

Models (in order of interpretability)

- Logistic regression, Discriminant analysis, Trees, Random Forests, Bagging, Boosting, SVM, etc.

Logistic Regression

Binary logistic regression (BLR)

Model to predict probability that Y is a “success”

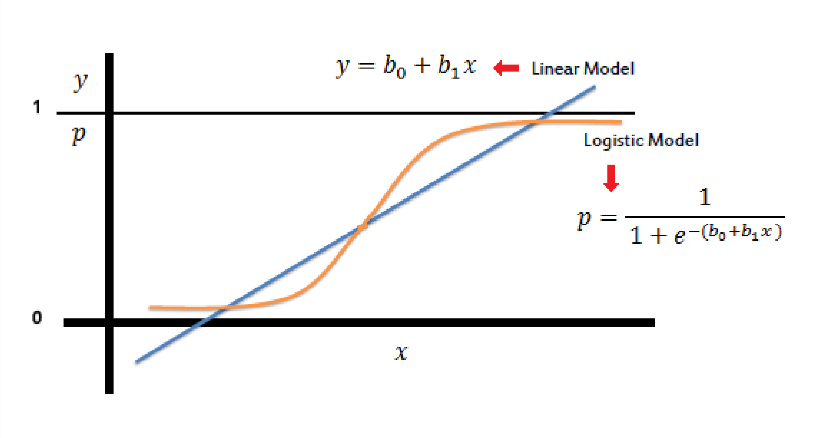
$$\text{logit}(success) = \beta_0 + \beta_1 x_1 + ...$$

$$odds(success) = e^{\beta_0 + \beta_1 x_1 + ...}$$

$$P(success) = \frac{e^{\beta_0 + \beta_1 x_1 + ...}}{1 + e^{\beta_0 + \beta_1 x_1 + ...}} = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + ...)}}$$

Logistic vs linear regression

- estimates of logistic model in interval (0, 1)
- estimates of linear model are not probabilities



Link function

Logistic model is a *generalized linear model* with the logit link function

$$\text{logit}(success) = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k$$

The "logit" is the log of the odds, so that after exponentiation

$$odds(success) = e^{\beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k}$$

and the relationship between odds and probabilities is

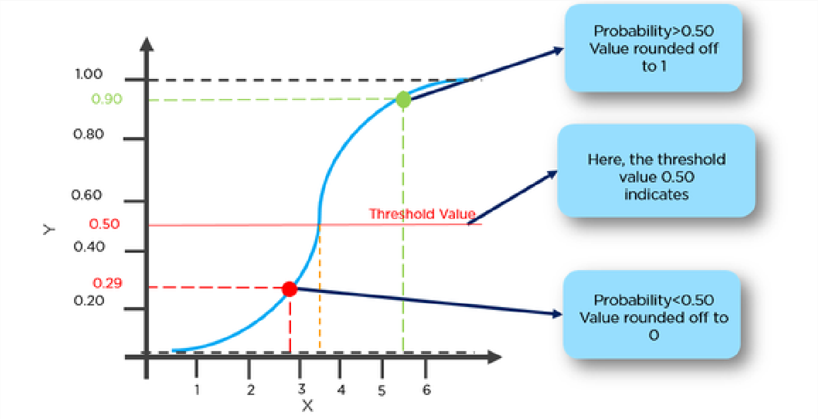
$$P(success) = \frac{odds(success)}{1 + odds(success)}$$

Classification procedure in R

Conversion of estimated probabilities into classifications

```
fit_glm  <- glm(y ~ x1 + x2 . . .,
               family = "binomial",
               link = "logit",        # default for "binomial"
               data   = <data>)

pred_glm  <- predict(fit_glm,
                    data = <data>,
                    type = "response")  # alternative is "link"

class_glm <- factor(pred_glm > 0.5,
                   labels = c("success", "failure"))
```

Schematically



Example

Logit of diabetes by females of Pima tribe.

$$\text{logit}(diabetes) = -9.514 + 0.141 \cdot npreg + 0.037 \cdot glu \;+\; ...$$

```
pima_glm <- glm(type ~ ., binomial, Pima.te)
coef(summary(pima_glm)) %>% round(3)
```

```
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   -9.514      1.229  -7.740    0.000
npreg          0.141      0.060   2.363    0.018
glu            0.037      0.006   6.743    0.000
bp            -0.009      0.013  -0.689    0.491
skin           0.013      0.020   0.658    0.511
bmi            0.079      0.028   2.777    0.005
ped            1.110      0.447   2.484    0.013
age            0.018      0.018   0.983    0.325
```

Classification

Probability estimates and classifications of first 10 cases:

```
p <- predict(pima_glm, Pima.te, type = "response")
data.frame(
  probability    = round(p[1:10], 3),
  classification = factor(p[1:10] > .5, labels = c("no diabetes", "diabetes")))
```

```
   probability classification
1        0.724       diabetes
2        0.035    no diabetes
3        0.029    no diabetes
4        0.048    no diabetes
5        0.843       diabetes
6        0.656       diabetes
7        0.398    no diabetes
8        0.258    no diabetes
9        0.445    no diabetes
10       0.286    no diabetes
```

Pros and cons

Pros

- straightforward interpretation of effects of predictors
- weak assumptions w.r.t. distribution of features

Cons

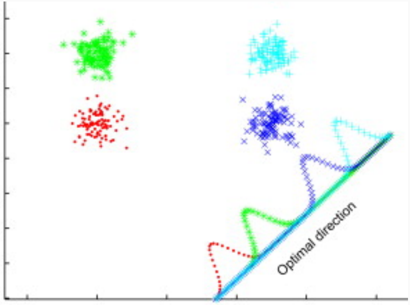
- unreliable parameter estimates when
    - large number of predictors
    - predictors with rare categories

Discriminant Analysis

What's discriminant analysis

Separates classes based on k discriminant functions

- k directions in feature space that best separate between classes
- $k = \text{min}(\#classes - 1, \#features - 1)$



Linear Discriminant Analysis (LDA)

Estimate *posterior* probability $P(X = x|Y = j)$ of class $j = 1, \dots, J$

$$P(Y = j|X = x) = \frac{\pi_j P(X = x|Y = j)}{\sum_{k=1}^{J} \pi_k P(X = x|Y = k)}$$

- π_j is *prior probability* of class j (sample proportion)
- $P(X = x|Y = j)$ are sample means of X within classes of Y

Linear discriminant functions

Linear discriminant functions

$$LD_j = c_{1j}X_1 + \cdots + c_{pj}X_p$$

- LD_1 separates the classes best, LD_2 second best, and so on
- LD's are orthogonal

Assumption $X|Y \sim N(\mu, \Sigma)$

- X is multivariate normal within each class
- X has covariance matrix Σ within each class

Quadratic Discriminant Analysis (QDA)

Estimates covariance matrix Σ_j for each class, with

- quadratic discriminant functions
- more parameters, so less bias but higher variance

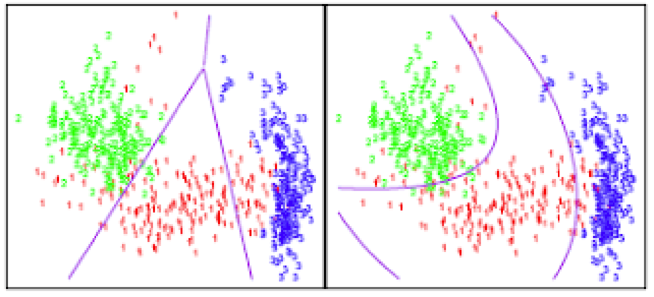


Figure 1: Linear vs quadratic discriminant functions

DA in R

Functions `lda()` and `qda()` of base R package `MASS`

- for LDA (for QDA it works the same):

```
fit_lda   <- lda(formula, data = <data>)

pred_lda  <- predict(fit_lda, newdata = <data>)

prob_lda  <- pred_lda$posterior

class_lda  <- pred_lda$class
```

Example

```
pima_lda <- lda(type ~ ., Pima.te)
```

- prior probabilities π

```
pima_lda$prior %>% round(3)
```

```
   No   Yes 
0.672 0.328 
```

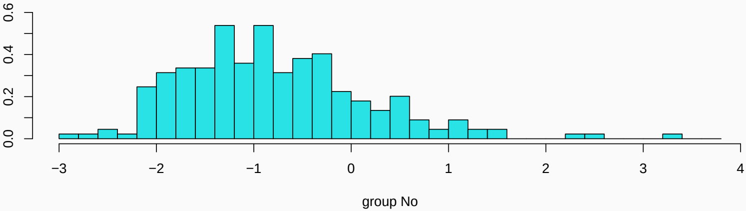
- conditional means $P(X = x|Y = j)$

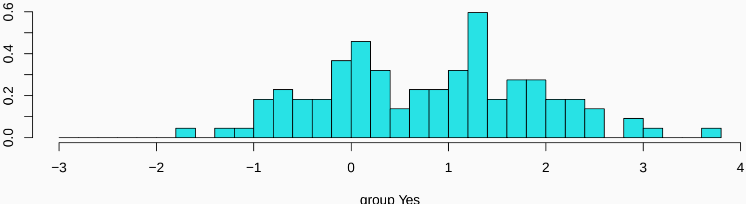
```
pima_lda$means %>% round(1)
```

```
    npreg   glu   bp skin  bmi ped  age
No    2.9 108.2 70.1 27.3 31.6 0.5 29.2
Yes   4.6 141.9 74.8 32.9 36.5 0.7 35.6
```

Linear discriminant function and projections

	npreg	glu	bp	skin	bmi	ped	age
LD1	0.1	0.0284	-0.0046	0.0047	0.052	0.6157	0.0122





Predictions

```r
pima_pred <- predict(pima_lda, Pima.te)
```

- first 10 predictions

```r
data.frame(
  posterior = round(pima_pred$posterior, 3),
  class     = pima_pred$class
)[1:10, ]
```

```
   posterior.No posterior.Yes class
1         0.255         0.745   Yes
2         0.977         0.023    No
3         0.980         0.020    No
4         0.969         0.031    No
5         0.112         0.888   Yes
6         0.281         0.719   Yes
7         0.659         0.341    No
8         0.798         0.202    No
9         0.590         0.410    No
10        0.722         0.278    No
```

Pros and cons

Pros

- performs better in conditions where logistic regression is unstable

Cons

- LDA depends on normality assumptions and equality of covariance matrices
- QDA relaxed equality assumption but is more complex (high variance)

Classification criteria

Goodness-of-fit criteria

a. Deviance statistic (the closer, to 0 the better the fit)

$$D = 2\sum_i y_i \log \frac{y_i}{\hat{\pi}_i} + (1 - y_i) \log \frac{1 - y_i}{1 - \hat{\pi}_i}$$

b. AIC (the smaller the value, the better the fit)
   - deviance plus penalty for model complexity (2 times # parameters)

c. Confusion matrix (accuracy of classifications)
   - proportions correctly/incorrectly classified

d. ROC curve
   - sensitivity and specificity for sequence of cut-off values
   - Area Under Curve (AUC) (50% is guessing, 100% is perfect)

Deviance and AIC

- Model with 1 predictor

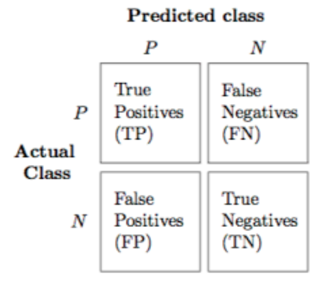
```
Call:  glm(formula = type ~ glu, family = binomial, data = Pima.te)

Coefficients:
(Intercept)          glu
   -5.94681      0.04242

Degrees of Freedom: 331 Total (i.e. Null);  330 Residual
Null Deviance:      420.3
Residual Deviance: 326  AIC: 330
```

- Model with all predictors

```
                     value
Residual Deviance 285.7914
AIC               301.7914
```

Confusion matrix

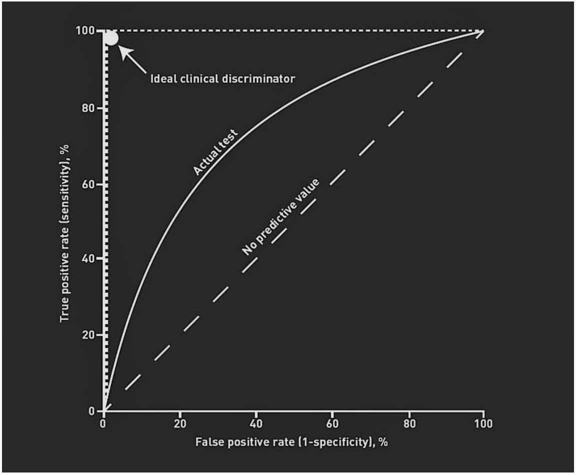
- Accuracy: $(TP + TN)/(TP + TN + FP + FN)$
- Misclassification error rate = 1 - accuracy
- Sensitivity: $TP/(TP + FN)$
- Specificity: $TN/(TN + FP)$



Different cutoff values result in different matrices

ROC and AUC

- the larger the area under the curve, the better the model performance



Model comparisons

Cross-validate accuracy

```
fit_cv <- train(type ~ .,
                data      = Pima.te,
                method    = "glm",
                metric    = "Accuracy",
                trControl = trainControl(method = "cv",
                                         number = 5))
```

Compare accuracy of Pima.te for `glm`, `lda` and `qda`:

- with cross-validation: `fit_cv$results$Accuracy`
- without cross-validation: `fit_cv$finalModel`

Confusion matrices final models

GLM

```
        estimated
observed  No Yes
     No  201  22
     Yes  46  63
```

LDA

```
        estimated
observed  No Yes
     No  199  24
     Yes  47  62
```

QDA

```
        estimated
observed  No Yes
     No  199  24
     Yes  40  69
```

Accuracy

Accuracy with and without cross-validation:

- GLM performs best with cross-validation
- QDA performs best without cross-validation

	Cross-validated	Final model
BLR	0.783	0.795
LDA	0.780	0.786
QDA	0.786	0.807

ROC's and AUC's