Slide 1

Reference: Agresti, Sections 4.6, 5.2.4, 11.1.

- Model selection is an important task in data analysis.
- The process of selecting a subset of variables from a large number of candidates is particularly important.
- A model with a large number of predictors can overfit the data and have poor predictive performance on new data.
- With p covariates, there are 2^p possible models.
- There are several automatic variable selection methods: forward selection, backward selection, stepwise selection.
- The change in deviance (*i.e.*, likelihood ratio test) can be used to compare the fit of two nested models.

Slide 2

Automatic selection procedures

- The three commonly used automatic model selection techniques are: forward selection, backward elimination, and stepwise selection.
- It is not guaranteed that the final model selected by each of these procedures would be the same.
- These procedures build models in a sequential manner by specifying selection and stopping criteria, which may be based on the likelihood ratio statistics, such as AIC.

Slide 3

Information criteria

- A commonly used model selection criteria is the **Akaike information criterion** (AIC):

$$AIC = -2\log L(\hat{\beta}) + 2p,$$

  where p is the number of parameters in the model.

- The criterion penalizes for model complexity.
- Another criterion with stronger penalty is the **Bayesian information criterion** (BIC):

$$BIC = -2\log L(\hat{\beta}) + p\log(n)$$

- The preferred model is the one with lowest AIC or BIC value.
- These criteria do not require the models being compared to be nested.

Slide 4

Testing nested models

- Suppose we want to compare the fit of two nested models, $\omega_0 \subset \omega_1$
- The test statistic is

$$\Delta G^2 = G_0^2 - G_1^2 \quad \xrightarrow{d} \chi^2_\nu,$$

  where G_0^2 is the deviance for ω_0, G_1^2 is the deviance for ω_1, and ν is the difference in the number of parameters between the two models.

- An asymptotically equivalent test is based on the Pearson statistics, $\Delta\mathcal{X}^2 = \mathcal{X}_0^2 - \mathcal{X}_1^2$.

Slide 5

Analysis of deviance

- In ordinary linear models, we compare models by examining the change in residual sum of squares as new predictors are added.
- In analysis of deviance, model selection is based on the change in deviance

Model	G^2	df
Saturated	0	0
Maximal	$G^2_{\max}$	$df_{\max}$
Model A	G^2_A	df_A
Model B	G^2_B	df_B
$\vdots$	$\vdots$	$\vdots$
Null	G^2_{null}	df_{null}

Slide 6

Example

Consider the following $2 \times 2 \times 3$ table that classifies 800 boys according to S = socioeconomic status, B = boy scout, and D = juvenile delinquency:

Socio-economic status	Boy scout	Delinquent yes	Delinquent no
Low	Yes	11	43
	No	42	169
Medium	Yes	14	104
	No	20	132
High	Yes	8	196
	No	2	59

Slide 7

The data can be re-arranged in this form:

S	B	y_i	n_i
low	scout	11	54
low	non-scout	42	211
medium	scout	14	118
medium	non-scout	20	152
high	scout	8	204
high	non-scout	2	61

Slide 8

Let us fit all possible models:

```
delinquent = data.frame(ses = as.factor(rep(c("low", "medium", "high"),rep(2,3))),
         boy = as.factor(rep(c("scout", "nonscout"),3)),
         y = c(11, 42, 14, 20, 8, 2), n = c(54, 211, 118, 152, 204, 61))

# R uses the first levels as reference.
# To use boy="non-scouts" and ses="low" as reference
delinquent$boy = relevel(delinquent$boy, ref="nonscout")
delinquent$ses = relevel(delinquent$ses, ref="low")

fit.null = glm(y/n ~ 1, weights=n, family=binomial, data=delinquent)
fit.boy = glm(y/n ~ boy, weights=n, family=binomial, data=delinquent)
fit.ses = glm(y/n ~ ses, weights=n, family=binomial, data=delinquent)
fit.boyses = glm(y/n ~ ses+boy, weights=n, family=binomial, data=delinquent)
fit.saturate = glm(y/n ~ boy*ses, weights=n, family=binomial, data=delinquent)
```

Slide 9

Let us collect the deviance statistics for all the models:

Model	G^2	df	p
Saturated	0.000	0	–
$S + B$	0.154	2	0.926
S	0.162	3	0.983
B	28.802	4	8.6×10^{-6}
Null (intercept only)	36.415	5	7.85×10^{-7}

Slide 10

Comparing nested models:

H_0 : smaller model fits well

	ΔG	Δdf	p-value
null vs. B	7.613	1	0.006
null vs. S	36.252	2	1.3×10^{-8}
B vs. $S + B$	28.648	2	6.0×10^{-7}
S vs. $S + B$	0.008	1	0.929
$S + B$ vs. $S * B$	0.154	2	0.926

Therefore, we should choose the S model, the simplest one that fits the data well.

Slide 11

Let's now use AIC and BIC to compare the models:

Model	AIC	BIC
Saturated	36.924	35.675
$S + B$	33.078	32.245
S	31.086	30.462
B	58.726	57.310
Null (intercept only)	63.339	63.131

AIC and BIC would also lead to the selection of the S model as the best fitting parsimonious model.

Slide 12

Let's now use the stepwise selection:

```
> step(fit.null, scope=list(lower=fit.null, upper=fit.saturate), direction="both")
Start:  AIC=63.34
y/n ~ 1

       Df Deviance    AIC
+ ses   2    0.162 31.086
+ boy   1   28.802 57.726
<none>      36.415 63.339

Step:  AIC=31.09
y/n ~ ses

       Df Deviance    AIC
<none>       0.162 31.086
+ boy   1    0.154 33.078
```

Slide 13

```
- ses   2   36.415 63.339

Call:  glm(formula = y/n ~ ses, family = binomial, data = delinquent,
    weights = n)

Coefficients:
(Intercept)      seshigh    sesmedium
    -1.3863      -1.8524      -0.5512

Degrees of Freedom: 5 Total (i.e. Null);  3 Residual
Null Deviance:     36.41
Residual Deviance: 0.1623  AIC: 31.09
```

Slide 14

Variable selection with high-dimensional data

- The methods discussed above cannot handle very large p.
- With binary outcomes, complete or quasi-complete separation often occurs resulting in some infinite estimates.
- Even when finite estimates exist, they may be imprecise because of ill-conditioning of the covariance matrix.

Slide 15

Penalized likelihood methods

- In fitting GLMs, **regularization methods** modify ML to give sensible answers in unstable situations.
- For a model with log-likelihood function, $l(\boldsymbol{\beta})$, we maximize

$$l^*(\boldsymbol{\beta}) = l(\boldsymbol{\beta}) - \eta(\boldsymbol{\beta}),$$

  where $\eta(.)$ is a penalty function.
- A variety of penalized-likelihood methods use the L_q**-norm** penalty

$$\eta(\boldsymbol{\beta}) = \lambda \sum_{j=1}^{p} |\beta_j|^q$$

  for some $q \geq 0$ and $\lambda \geq 0$.
- λ is called a **smoothing parameter** and its choice reflects the bias-variance tradeoff: increasing λ results in greater shrinkage and smaller variance but greater bias.

Slide 16

λ is usually chosen by k-fold cross-validation:

- the data is divided into k subsets; $k-1$ of the subsets are used as training and the remaining subset as validation
- for each λ value over a grid, the model is fit to the training set and its prediction is assessed on the validation set
- this is repeated using each of the k subsets as validation
- the λ value having the lowest sample mean prediction error is selected.

Slide 17

The lasso: L_1-norm penalty

- The **lasso** (least absolute shrinkage and selection operator) uses the L_1-norm penalty (Tibshirani, 1996)

$$\eta(\boldsymbol{\beta}) = \lambda \sum_j |\beta_j|.$$

- For λ sufficiently large, this method shrinks some $\hat{\beta}_j$ completely to 0 and can be used for variable selection.
- There are various optimization methods for estimating $\boldsymbol{\beta}$ subject to the penalty (least angle regression - LARS, coordinate descent, etc).
- Various regularization methods have been proposed to overcome some of the limitations of lasso (elastic net, adaptive lasso, smoothly clipped absolute deviation-SCAD).

Slide 18

Let's consider the *Pima Indian Diabetes* data from the UCI Machine Learning Repository, available in the R package `mlbench`. The data contain the following variables on 768 adult female patients ($\geq$ 21 yeard old) of Pima Indian heritage:

pregnant	Number of times pregnant
glucose	Plasma glucose concentration after oral glucose tolerance test
pressure	Diastolic blood pressure (mm Hg)
triceps	Triceps skin fold thickness (mm)
insulin	2-Hour serum insulin (μU/ml)
mass	Body mass index (weight in kg/(height in m)^2)
pedigree	Diabetes pedigree function ("synthesis of diabetes mellitus history in relatives and genetic relationship of those relatives to subject")
age	Age (years)
diabetes	Class variable (0/1 for diabetes)

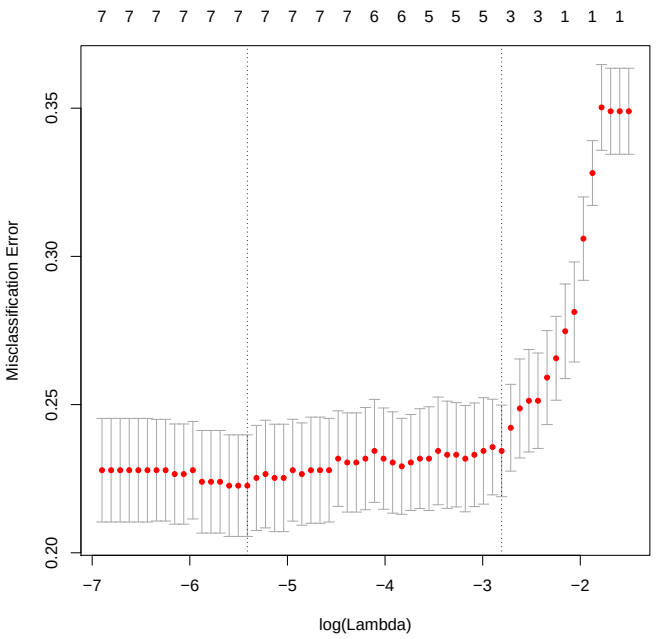
Slide 19

```
library(glmnet)

data("PimaIndiansDiabetes")
# We can use the function model.matrix to create the design matrix
# converting categorical predictors to appropriate dummy variables
X = model.matrix(diabetes ~ ., data=PimaIndiansDiabetes)
Y = as.numeric(PimaIndiansDiabetes$diabetes=="pos")

# cv.glmnet is the main function to do cross-validation.
# Here we use it with the misclassification error as criterion.
# Other options include "deviance" and "auc"
cvfit = cv.glmnet(x=X[,-1], y=Y, family="binomial", type.measure="class")
plot(cvfit)
```

Slide 20



Slide 21

```
#lambda.min is the value of lambda that gives minimum mean cross-validated error.
lambda_min = cvfit$lambda.min
> lambda_min
[1] 0.004468368

# The other lambda saved is lambda.1se, which gives
# the most regularized model such that error is within one standard error
# model with smallest number of coefficients that also gives a good accuracy
lambda_1se = cvfit$lambda.1se
> lambda_1se
[1] 0.06045915
```

Slide 22

```
#regression coefficients
> coef(cvfit, s=lambda_1se)
9 x 1 sparse Matrix of class "dgCMatrix"
                      1
(Intercept) -4.52844463
pregnant     0.03643469
glucose      0.02264484
pressure     .
triceps      .
insulin      .
mass         0.02943169
pedigree     .
age          .
```

Slide 23

```
# mean cross-validated error
pred.err = cvfit$cvm[cvfit$lambda==lambda_1se]
> pred.err
[1] 0.2395833
```

Slide 24

Assessing predictive power

- The proportion of samples classified correctly is a good criterion to assess predictive power.
- Validation techniques are usually divided into two types:
  - in external validation the model is applied to a completely new data derived from the same source
  - internal validation uses the same data for model building and its assessment – this gives an optimistic evaluation.

  A compromise is to use a cross-validation approach.

Slide 25

Receiving operating characteristic (ROC) curve

- The predictive power can be summarized by:

  sensitivity $= p(\hat{y}_i = 1 | y_i = 1)$,

  specificity $= p(\hat{y}_i = 0 | y_i = 0)$,

  where $\hat{y}_i = 1$ if $p(\hat{\pi}_i > \pi_0)$; and $\hat{y}_i = 0$ otherwise.
- There is an inherent trade-off between sensitivity and specificity.
- A ROC curve is a plot of sensitivity as a function of (1-specificity) for different cutoffs π_0.
- The higher the area under the curve (AUC), the better the predictions.

Slide 26

```
pima.fit = glm(diabetes ~ pregnant+glucose+mass,
      family=binomial, data=PimaIndiansDiabetes)

> summary(pima.fit)
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -8.124024   0.638486 -12.724  < 2e-16 ***
pregnant     0.137094   0.026768   5.121 3.03e-07 ***
glucose      0.034162   0.003312  10.316  < 2e-16 ***
mass         0.081551   0.013736   5.937 2.90e-09 ***

    Null deviance: 993.48  on 767  degrees of freedom
Residual deviance: 744.12  on 764  degrees of freedom
AIC: 752.12
```

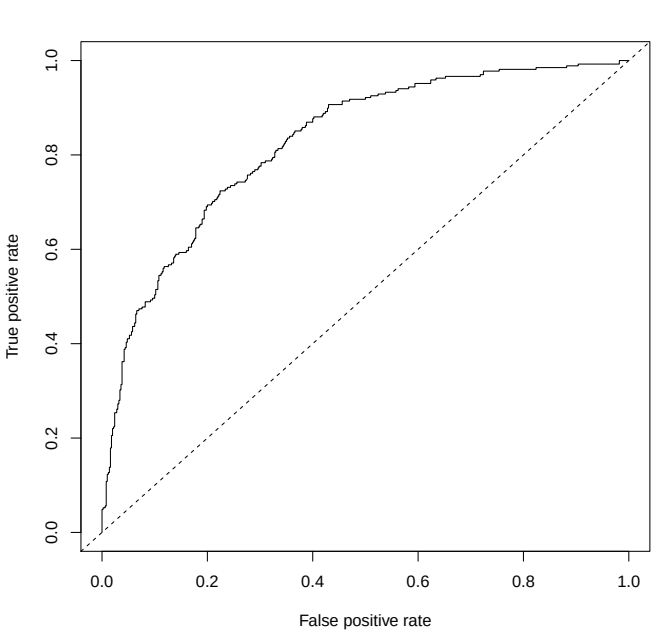
Slide 27

```
library(ROCR)

# Plot ROC curve
pred = prediction(fitted(pima.fit), PimaIndiansDiabetes$diabetes)
perf = performance(pred, "tpr", "fpr")
plot(perf)
abline(a=0, b=1, lty=2)

# Area under ROC curve (AUC) = concordance index
auc.perf = performance(pred, "auc")
auc.perf@y.values
> auc.perf@y.values
[[1]]
[1] 0.8260485
```

Slide 28



Slide 29

R^2 measures

- In ordinary regression, the *coefficient of determination*, R^2, describes the proportion of the variability in Y explained by the linear regression of Y on X

$$R^2 = \frac{SST - SSE}{SST} = \frac{G_0^2 - G_m^2}{G_0^2},$$

  where G_0^2 is the deviance of the null (intercept-only) model and G_m^2 is the deviance of the fitted model.

- This is equivalent to the measure based on the maximized log-likelihoods (Agresti, p. 147)

$$\frac{L_M - L_0}{L_S - L_0},$$

  where L_M is the maximized log-likelihood for a given model, L_S for the saturated model, and L_0 for the null model.

Slide 30

```
> (pima.fit$null.deviance-pima.fit$deviance)/pima.fit$null.deviance
[1] 0.2509945
```

- For any GLM, the correlation between the observed response y_i and the fitted values $\hat{\mu}_i$ can be used to measure predictive power.

```
> cor(Y, fitted(pima.fit))
[1] 0.5500234
```

- These measures are mainly useful for comparing models.

Slide 31

Caveats/Considerations

- Statistical significance is not the same as practical significance, so a significance test should not be the sole criterion for including a variable in a model.
- Many models can be consistent with the data.
- A model that fits the data may not necessarily predict well, since this depends on how predictable the outcome is.

Slide 32

Confounding

- We want to control for covariates that can influence the relationship between X and Y.
- We can adjust for confounding in the study design (by random assignment or matching) or in the analysis (by stratification or modeling).
- Omitting confounding variables can:
  - mask the effect of an important variable;
  - induce an effect where none exists;
  - reverse the direction of an association (Simpson's paradox).

Slide 33

Interaction

Let's consider again the low birth data `BirthWeight.txt`. The variables in the data are:

ID	Identification code
LOW	Low birth weight (0=weight>2500g, 1=weight<2500g)
AGE	Age of mother in years
LWT	Weight in pounds at last menstrual period
RACE	Race (1=White, 2=Black, 3=Other)
SMOKE	Smoking status during pregnancy (1=Yes, 0=No)
PTL	History of premature labor (0=None, 1=One, 2=Two, etc.)
HT	History of hypertension (1=Yes, 0=No)
UI	Presence of uterine irritability (1=Yes, 0=No)
FTV	Number of physician visits during the first trimester
BWT	Birth weight in grams

Slide 34

- Let's fit a logistic regression model on mother's pre-pregnancy weight and smoking status.
- The additive/main effects model is

$$\text{logit}(\pi_i) = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2},$$

$$X_{i1} = \text{mother's weight} \quad X_{i2} = \begin{cases} 1 & \text{for smokers} \\ 0 & \text{otherwise} \end{cases}.$$

This model implies that the association between mother's weight and the risk of having a low weight infant is the same regardless of mother's smoking status.

Slide 35

- When interaction is present, the effect of mother's smoking on her risk of having a low birth weight infant is modified by mother's weight

$$\text{logit}(\pi_i) = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \beta_3 X_{i1} X_{i2}.$$

- What do β_1, β_2 and β_3 measure?

Slide 36

```
bwt.main = glm(low ~ lwt + as.factor(smoke), family=binomial)

> summary(bwt.main)
Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)        0.62200    0.79592   0.781   0.4345
lwt               -0.01332    0.00609  -2.188   0.0287 *
as.factor(smoke)1  0.67667    0.32470   2.084   0.0372 *

    Null deviance: 234.67  on 188  degrees of freedom
Residual deviance: 224.34  on 186  degrees of freedom
AIC: 230.34
```

Slide 37

```
bwt.inter = glm(low ~ lwt * as.factor(smoke), family=binomial)

> summary(bwt.inter)
Coefficients:
                      Estimate Std. Error z value Pr(>|z|)
(Intercept)            1.93234    1.29209   1.496   0.1348
lwt                   -0.02389    0.01039  -2.299   0.0215 *
as.factor(smoke)1     -1.51089    1.61737  -0.934   0.3502
lwt:as.factor(smoke)1  0.01757    0.01279   1.373   0.1697

    Null deviance: 234.67  on 188  degrees of freedom
Residual deviance: 222.37  on 185  degrees of freedom
AIC: 230.37
```

Slide 38

Collinearity

- One possible consequence of fitting models with many covariates is the increasing chance of collinear relationships among the variables.
- Interaction terms are particularly prone to collinearity.
- Collinearity results in very large estimates of standard error and possibly very large estimated regression coefficients.
- If collinearity is particularly severe, it might not be possible to obtain parameter estimates (i.e., failure to converge).
- Centering may reduce the correlation and may eliminate collinearity problems for models with interaction terms.

Slide 39

```
BirthWeight$lwt.center = BirthWeight$lwt - mean(BirthWeight$lwt)
bwt.center = glm(low ~ lwt.center * as.factor(smoke),
       family=binomial, data=BirthWeight)

> summary(bwt.center)
Coefficients:
                              Estimate Std. Error z value Pr(>|z|)
(Intercept)                   -1.16847    0.23330  -5.008 5.49e-07 ***
lwt.center                    -0.02389    0.01039  -2.299   0.0215 *
as.factor(smoke)1              0.76970    0.33422   2.303   0.0213 *
lwt.center:as.factor(smoke)1   0.01757    0.01279   1.373   0.1697

    Null deviance: 234.67  on 188  degrees of freedom
Residual deviance: 222.37  on 185  degrees of freedom
AIC: 230.37
```