MATH 5910

Dimension Reduction

Dimension Reduction

- Recall
  - n observations
  - p variables
- Some methods will NOT work if $p > n$
- Even if $p < n$, may want to select/reduce variables
- Need to reduce the dimension of variable p
- Possible remedies
  - Shrinkage/regularization methods (ridge, LASSO, elastic net)
  - Principal component analysis (PCA)

Regression

Recall

- RSS (for regression, slight changes in notation)

$$\text{RSS} = \sum_{i=1}^{n} \left(Y_i - \beta_0 - \boldsymbol{x}_i'\boldsymbol{\beta}\right)^2$$

  for $\boldsymbol{x}_i' = (x_{i1}, \ldots, x_{ip})$ and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)'$
- Minimize RSS to obtain $\hat{\beta}_0$ and $\hat{\boldsymbol{\beta}}$
- Ridge regression: Adjust RSS for numerical stability, can handle $p > n$ case

Ridge Regression

Ridge regression

- Find β_0 and $\boldsymbol{\beta}$ that minimizes

$$\text{RSS} + \lambda \sum_{j=1}^{p} \beta_j^2 = \text{RSS} + \lambda \|\boldsymbol{\beta}\|_2^2$$

where

$$\|\boldsymbol{\beta}\|_2 = \sqrt{\sum_{j=1}^{p} \beta_j^2}$$

- Added the penalty term $\lambda \|\boldsymbol{\beta}\|_2^2$

Ridge Regression

- Recall for regression

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y}$$

- For ridge regression, it can be seen that

$$\hat{\boldsymbol{\beta}}_{ridge} = (\boldsymbol{X}'\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}'\boldsymbol{Y}$$

Ridge Regression

What about λ?

- Must be estimated
- Should be data driven
- Popular method for estimating λ: cross-validation (CV)

LASSO

Least Absolute Shrinkage and Selection Operator (LASSO)

- Similar to ridge regression
- Difference: use $\|\boldsymbol{\beta}\|_1$ in place of $\|\boldsymbol{\beta}\|_2^2$, where

$$\|\boldsymbol{\beta}\|_1 = \sum_{j=1}^{p} |\beta_j|$$

LASSO

Find β_0 and $\boldsymbol{\beta}$ that minimizes

$$\text{RSS} + \lambda \|\boldsymbol{\beta}\|_1$$

Again, λ estimated by CV

LASSO

- Does not have a closed-form solution like ridge regression
- The β_0 and $\boldsymbol{\beta}$ must be estimated numerically

LASSO

- The objective function

$$\text{RSS} + \lambda \|\boldsymbol{\beta}\|_1$$

can be reformulated as: minimize RSS, subject to

$$\|\boldsymbol{\beta}\|_1 = \sum_{j=1}^{p} |\beta_j| \leq s$$

- This forces some β_j to be zero
- So, LASSO naturally performs variable selection

Elastic Net

Combination of ridge and LASSO

- Objective function

$$\text{RSS} + \lambda[(1-\alpha)\|\boldsymbol{\beta}\|_2^2 + \alpha\|\boldsymbol{\beta}\|_1]$$

- If $\alpha = 1$, then LASSO
- If $\alpha = 0$, then ridge regression
- May choose α between 0 and 1

Implementation in R

Use package `glmnet`

- Main functions: `glmnet()` and `cv.glmnet()`
- Can be used for regression, logistic regression, others (similar to `glm()`)

Example 1

Introductory example, from the website

- Glmnet Website

  `https://glmnet.stanford.edu/`

- Resources for the `glmnet` package

Example 1

Regression example

```
# Load glmnet package

library(glmnet)

# Extract data

data(QuickStartExample)
x <- QuickStartExample$x
y <- QuickStartExample$y
dim(x)
```

Example 1

LASSO (`glmnet()` defaults to LASSO, `alpha=1`)

```
fit <- glmnet(x, y)
plot(fit)
print(fit)
```

Coefficients with $\lambda = 0.1$

```
coef(fit,s=0.1)
```

Prediction with multiple λ

```
set.seed(29)
nx <- matrix(rnorm(5 * 20), 5, 20)
predict(fit, newx = nx, s = c(0.1, 0.05))
```

Example 1

CV

```
cvfit <- cv.glmnet(x, y)
plot(cvfit)
cvfit$lambda.min
coef(cvfit, s = "lambda.min")
predict(cvfit, newx = x[1:5,], s = "lambda.min")
```

Note the selected variables from coef

Example 1

To regress on selected variables, may try

```
coef(cvfit, s = "lambda.min")!=0
(coef(cvfit, s = "lambda.min")!=0)[-1]
summary(lm(y~x[,(coef(cvfit, s = "lambda.min")!=0)[-1]]))

as.numeric(coef(lm(y~x[,(coef(cvfit, s = "lambda.min")!=0)[-1]])))
as.numeric(coef(cvfit, s = "lambda.min"))
```

Example 1

Recall

```
plot(cvfit)
```

Can take a look at the other "default" lambda value

```
cvfit$lambda.1se
coef(cvfit, s = "lambda.1se")
summary(lm(y~x[,(coef(cvfit, s = "lambda.1se")!=0)[-1]]))

as.numeric(coef(lm(y~x[,(coef(cvfit, s = "lambda.1se")!=0)[-1]])))
as.numeric(coef(cvfit, s = "lambda.1se"))
```

Example 2

Ridge regression

- Take `alpha=0` in `glmnet()` and `cv.glmnet()`
- Reminder: Does NOT perform variable selection

Example 2

In R

```
fit0 <- glmnet(x, y, alpha=0)
plot(fit0)
plot(fit0, xvar = "lambda", label = TRUE)
print(fit0)
```

Example 2

CV

```
cvfit0 <- cv.glmnet(x, y, alpha=0)
plot(cvfit0)
cvfit0$lambda.min
coef(cvfit0, s = "lambda.min")
predict(cvfit0, newx = x[1:5,], s = "lambda.min")
```

Example 3

Regression, elastic net with weights

```
wts <- c(rep(1,50), rep(2,50))
fit <- glmnet(x, y, alpha = 0.2, weights = wts, nlambda = 20)

print(fit)
```

Example 4

Compare coefficient, exact versus approximate, then predict

```
fit <- glmnet(x, y)
any(fit$lambda == 0.5)  # 0.5 not in original lambda sequence

coef.apprx <- coef(fit, s = 0.5, exact = FALSE)
coef.exact <- coef(fit, s = 0.5, exact = TRUE, x=x, y=y)
cbind2(coef.exact[which(coef.exact != 0)],
       coef.apprx[which(coef.apprx != 0)])

cbind2(coef.exact, coef.apprx)

predict(fit, newx = x[1:5,], type = "response", s = 0.05)
```

(Note the added inputs for exact=TRUE)

Example 4

Plotting options

```
plot(fit, xvar = "lambda", label = TRUE)

plot(fit, xvar = "dev", label = TRUE)
```

Example 5

More CV

```
cvfit <- cv.glmnet(x, y, type.measure = "mse", nfolds = 20)
print(cvfit)

cvfit$lambda.min
predict(cvfit, newx = x[1:5,], s = "lambda.min")
coef(cvfit, s = "lambda.min")
```

Example 5

Fine tune CV

```
foldid <- sample(1:10, size = length(y), replace = TRUE)
cv1  <- cv.glmnet(x, y, foldid = foldid, alpha = 1)
cv.5 <- cv.glmnet(x, y, foldid = foldid, alpha = 0.5)
cv0  <- cv.glmnet(x, y, foldid = foldid, alpha = 0)

par(mfrow = c(2,2))
plot(cv1); plot(cv.5); plot(cv0)
plot(log(cv1$lambda)   , cv1$cvm , pch = 19, col = "red",
     xlab = "log(Lambda)", ylab = cv1$name)
points(log(cv.5$lambda), cv.5$cvm, pch = 19, col = "grey")
points(log(cv0$lambda) , cv0$cvm , pch = 19, col = "blue")
legend("topleft", legend = c("alpha= 1", "alpha= .5", "alpha 0"),
       pch = 19, col = c("red","grey","blue"))
```

Example 6

More options

```
tfit <- glmnet(x, y, lower.limits = -0.7, upper.limits = 0.5)
plot(tfit)

p.fac <- rep(1, 20)
p.fac[c(1, 3, 5)] <- 0
pfit <- glmnet(x, y, penalty.factor = p.fac)
plot(pfit, label = TRUE)
```

Example 7

Logistic regression

- LASSO example
- Minor adjustments, but some important differences

Example 7

Data

```
data(BinomialExample)
x <- BinomialExample$x
y <- BinomialExample$y
dim(x)
```

Example 7

Fit logistic regression with LASSO

```
fit <- glmnet(x, y, family = "binomial")

predict(fit, newx = x[1:5,], type = "class", s = c(0.05, 0.01))
```

Example 7

CV, note different options

```
cvfit <- cv.glmnet(x, y, family = "binomial", type.measure = "class")
plot(cvfit)

cvfit$lambda.min
cvfit$lambda.1se
```

Example 7

Note that `type.measure = "class"` not needed if only interested in regression, can try others

```
cv.glmnet(x, y, family = "binomial")
cv.glmnet(x, y, family = "binomial", type.measure = "deviance")
cvfit
```

Example 8

Poisson regression, similar to logistic regression

```
data(PoissonExample)
x <- PoissonExample$x
y <- PoissonExample$y

fit <- glmnet(x, y, family = "poisson")
plot(fit)

coef(fit, s = 1)
predict(fit, newx = x[1:5,], type = "response", s = c(0.1,1))

cvfit <- cv.glmnet(x, y, family = "poisson")
```

PCA

Principal Component Analysis (PCA)

- Many derivations, many uses
- Focus here: dimension reduction
- Interpretation

PCA

Heuristic illustration

PCA

Starting point

- Estimated covariance (or correlation) matrix

$$\hat{\Sigma} = \boldsymbol{V}\boldsymbol{D}\boldsymbol{V}'$$

- $\boldsymbol{V} = [\phi_1, \phi_2, \ldots, \phi_p]$ matrix of eigenvectors
- $\boldsymbol{D}$ matrix, eigenvalues

$$\boldsymbol{D} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & & & \\ 0 & 0 & \cdots & \lambda_p \end{bmatrix}$$

PCA

- Principal components

$$
\begin{aligned}
Z_1 &= \phi_{11} X_1 + \phi_{21} X_2 + \cdots + \phi_{p1} X_p \\
Z_2 &= \phi_{12} X_1 + \phi_{22} X_2 + \cdots + \phi_{p2} X_p \\
&\vdots \\
&\vdots \\
Z_p &= \phi_{1p} X_1 + \phi_{2p} X_2 + \cdots + \phi_{pp} X_p
\end{aligned}
$$

where $\boldsymbol{\phi}_j = [\phi_{1j}, \phi_{2j}, \ldots, \phi_{pj}]'$ and $\boldsymbol{V} = [\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_p]$

PCA

- Choose the first M components ($M < n$):

$$
\begin{aligned}
Z_1 &= \phi_{11} X_1 + \phi_{21} X_2 + \cdots + \phi_{p1} X_p \\
Z_2 &= \phi_{12} X_1 + \phi_{22} X_2 + \cdots + \phi_{p2} X_p \\
&\vdots \\
Z_M &= \phi_{1M} X_1 + \phi_{2M} X_2 + \cdots + \phi_{pM} X_p
\end{aligned}
$$

- Reduces dimension from p to M, but still use all data
- $Z_1, \ldots, Z_M$: New (reduced) data

PCA

How many M?

- Eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_p$
- Percent variance explained (most common)
  - Find M such that

$$\frac{\lambda_1 + \lambda_2 + \cdots + \lambda_M}{\lambda_1 + \lambda_2 + \cdots + \lambda_{p-1} + \lambda_p}$$

    is over some prescribed percentage (80%, 90%, etc.)
- Can also use CV

PCA

Caveat

- Principal components not unique
- Eigenvectors $\phi_1, \phi_2, \ldots, \phi_p$ can have different signs
- Complicates interpretation

Example 9

PCA in R

- Function `prcomp()`
- Can also use `princomp()`
- Example from help file

Example 9

From ?prcomp

```
USArrests

prcomp(USArrests)  # inappropriate
prcomp(USArrests, scale = TRUE)
prcomp(~ Murder + Assault + Rape,
       data = USArrests, scale = TRUE)
plot(prcomp(USArrests))
summary(prcomp(USArrests, scale = TRUE))
biplot(prcomp(USArrests, scale = TRUE))
```

Example 9

The result from `biplot()`

- First 2 PCs
- Each state plotted
- PC scores (Z_1, Z_2)
- Can use X_1, X_2, X_3, X_4 from each state to create (Z_1, Z_2) for each state

$$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \phi_{31}X_3 + \phi_{41}X_4$$
$$Z_2 = \phi_{12}X_1 + \phi_{22}X_2 + \phi_{32}X_3 + \phi_{42}X_4$$

PCR

Principal Component Regression (PCR)

- Obtain $Z_1, Z_2, \ldots, Z_p$
- Choose M by CV or "percent variance explained"
- Regress Y on $Z_1, Z_2, \ldots, Z_M$

Advantages?

PCR

- Can use R packages (for example, `pls` package with `pcr()`)
- Example without packages

Example 10

```
longley
?longley

# A macroeconomic data set which provides a well-known example
# for a highly collinear regression.

longley.x <- data.matrix(longley[, 1:6])
longley.y <- longley[, "Employed"]
pairs(longley, main = "longley data")
summary(fm1 <- lm(Employed ~ ., data = longley))
```

Example 10

Can use variable selection

```
sfm1<-step(fm1)
summary(sfm1)
```

Example 10

May also try PCR

- First, perform PCA
- Determine M

```
prcomp(~ .-Employed, data = longley)
summary(prcomp(~ .-Employed, data = longley))
```

So only 2 components enough

Example 10

Can extract both ϕ's and Z's

```
prcomp(~ .-Employed, data = longley)$rotation
prcomp(~ .-Employed, data = longley)$x
```

Use the first M of Z's for PCR

Example 10

Use the first 2 PC

```
summary(lm(Employed~prcomp(~ .-Employed, data=longley)$x[,1:2],
           data = longley))

# Make it less messy
longley.pcx<-prcomp(~ .-Employed, data = longley)$x
summary(lm(Employed~longley.pcx[,1:2], data = longley))
```