CS6220: Data mining techniques
Multiple regression

Olga Vitek

October 1, 2015

Outline

Example of an arbitrary association

Confidence intervals vs prediction intervals

Multicollinearity

Multicollinearity and higher order terms

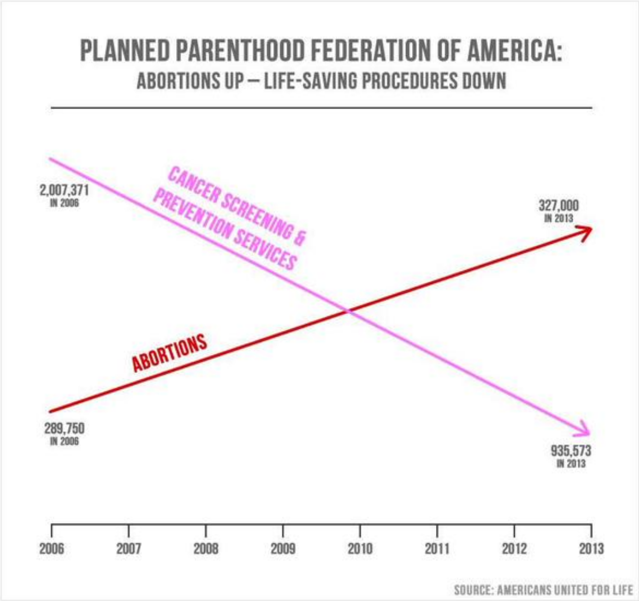
Steps of model building

Example: surgical unit

Evaluation of predictive ability

Example of an arbitrary association

Congressional hearing, October 29 2015



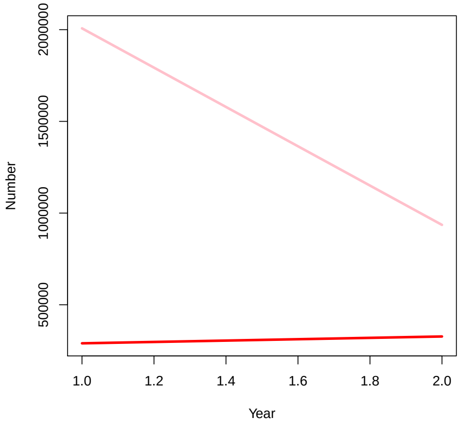
PLANNED PARENTHOOD FEDERATION OF AMERICA:

ABORTIONS UP – LIFE-SAVING PROCEDURES DOWN

SOURCE: AMERICANS UNITED FOR LIFE

Correct visualization

```
> pp <- data.frame(
+   screening=c(2007371, 935573), abortion=c(289750,327000))
> plot(1:2, c(min(pp), max(pp)), type='n', xlab='Year', ylab='Number')
> lines(1:2, pp$screening, type='l', col='pink', lwd=3)
> lines(1:2, pp$abortion, col='red', lwd=3)
```



Confidence intervals vs prediction intervals

Diamonds: a simple linear regression

```
> library(ggplot2)
> set.seed(123)
> index <- sample(1:nrow(diamonds), 50) # try a subset first
> diamonds2 <- diamonds[index,]
> fit <- lm(price ~ carat, data=diamonds2)
> fit

Call:
lm(formula = price ~ carat, data = diamonds2)

Coefficients:
(Intercept)        carat
      -2511         8060
```

Confidence intervals vs prediction intervals

```
> confint(fit)

                2.5 %    97.5 %
(Intercept) -3522.082 -1500.825
carat        6986.249  9134.442

> head(predict(fit, interval='confidence'))

            fit        lwr       upr
15512  5790.702  5287.9285  6293.476
42521  1518.719   933.4478  2103.990
22060 11513.547 10441.9885 12585.106
47628  1599.323  1020.6606  2177.985
50726  3130.788  2648.5774  3612.999
2458   3211.392  2732.3831  3690.400
```

Multicollinearity

Including correlated predictors is not helpful

```
> summary(lm(price ~ x, data=diamonds2))

Call:
lm(formula = price ~ x, data = diamonds2)

Residuals:
    Min      1Q  Median      3Q     Max
-2916.8 -1297.6  -120.8   874.9  6015.5

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   -14989       1524  -9.837 4.32e-13 ***
x               3280        256  12.812  < 2e-16 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.

Residual standard error: 1839 on 48 degrees of freedom
Multiple R-squared:  0.7737,	Adjusted R-squared:  0.769
F-statistic: 164.2 on 1 and 48 DF,  p-value: < 2.2e-16
```

Including correlated predictors is not helpful

```
> summary(lm(price ~ y, data=diamonds2))

Call:
lm(formula = price ~ y, data = diamonds2)

Residuals:
    Min      1Q  Median      3Q     Max 
-2737.3 -1396.8   -78.0   990.5  5811.7 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -14923.6     1512.2  -9.869 3.89e-13 ***
y             3272.1      254.3  12.867  < 2e-16 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.

Residual standard error: 1833 on 48 degrees of freedom
Multiple R-squared:  0.7753,        Adjusted R-squared:  0.7706 
F-statistic: 165.6 on 1 and 48 DF,  p-value: < 2.2e-16
```

Including correlated predictors is not helpful

```
> summary(lm(price ~ x+y, data=diamonds2))

Call:
lm(formula = price ~ x + y, data = diamonds2)

Residuals:
    Min      1Q  Median      3Q     Max
-2755.3 -1380.0   -71.8   977.9  5831.1

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   -14934       1538  -9.712 8.15e-13 ***
x                328       5240   0.063    0.950
y               2946       5222   0.564    0.575
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.

Residual standard error: 1852 on 47 degrees of freedom
Multiple R-squared:  0.7753,	Adjusted R-squared:  0.7657
F-statistic: 81.07 on 2 and 47 DF,  p-value: 5.808e-16
```

Multicollinearity and higher order terms

Carat is an important linear term

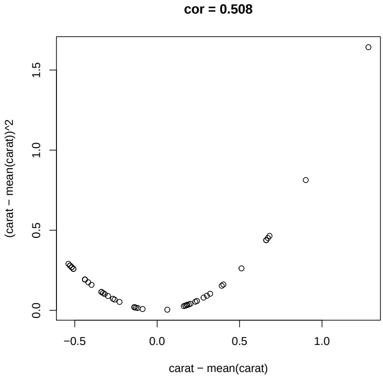
```
> summary(lm(price ~ carat, data=diamonds2))

Call:
lm(formula = price ~ carat, data = diamonds2)

Residuals:
    Min      1Q  Median      3Q     Max
-2803.9  -913.7   -20.2   583.3  5049.3

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -2511.5      502.6  -4.997 8.16e-06 ***
carat         8060.3      534.2  15.088  < 2e-16 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ

Residual standard error: 1613 on 48 degrees of freedom
Multiple R-squared:  0.8259,        Adjusted R-squared:  0.8222
F-statistic: 227.7 on 1 and 48 DF,  p-value: < 2.2e-16
```

But is much less important if we add a quadratic term

```
> summary(lm(price ~ carat + I(carat^2), data=diamonds2))

Call:
lm(formula = price ~ carat + I(carat^2), data = diamonds2)

Residuals:
    Min      1Q  Median      3Q     Max
-2763.4  -876.7    57.3   452.3  5012.7

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -2033.4      944.7  -2.152  0.03654 *
carat         6843.8     2099.9   3.259  0.00208 **
I(carat^2)     612.0     1021.2   0.599  0.55185
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ

Residual standard error: 1624 on 47 degrees of freedom
Multiple R-squared:  0.8272,        Adjusted R-squared:  0.8198
F-statistic: 112.5 on 2 and 47 DF,  p-value: < 2.2e-16
```

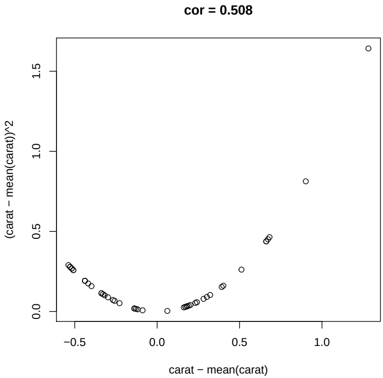
Carat and carat2 are correlated

```
> with(diamonds2, plot(carat, carat^2,
+  main=paste('cor =', round(cor(carat, carat^2), digits=3)))
+ )
```



Linear transofrmations remove some correlation

```
> with(diamonds2, plot(carat-mean(carat), (carat-mean(carat))^2,
+  main=paste('cor =',
+  round(cor(carat-mean(carat), (carat-mean(carat))^2), digits=3)))
+ )
```



Poly makes transformations that remove all correlation

```
> summary(lm(price ~ poly(carat, 2), data=diamonds2))

Call:
lm(formula = price ~ poly(carat, 2), data = diamonds2)

Residuals:
    Min      1Q  Median      3Q     Max
-2763.4  -876.7    57.3   452.3  5012.7

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)       4246.3      229.7  18.487   <2e-16 ***
poly(carat, 2)1  24341.5     1624.1  14.987   <2e-16 ***
poly(carat, 2)2    973.3     1624.1   0.599    0.552
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.

Residual standard error: 1624 on 47 degrees of freedom
Multiple R-squared:  0.8272,        Adjusted R-squared:  0.8198
F-statistic: 112.5 on 2 and 47 DF,  p-value: < 2.2e-16
```

Steps of model building

Steps of model building (1)

- Data examination
  - outliers? errors? missing data?
  - correct records; complete missings; remove unreliable predictors
- Preliminary model investigation
  - scatterplots; correlations between Xs and between Xs and Y; normality of errors
  - potential transformations of Y
  - remove redundant or uninformative variables
  - identify potentially important predictors that are not part of the dataset

Steps of model building (2)

- Further reduction of potential predictors: domain knowledge
- (Semi-)automated subset selection techniques
- Model refinement
  - higher-order terms (curvature, interactions)
  - consider influential or atypical observations
  - a small number of competing models can be kept at this stage
- Model validation
  - stability of estimated coefficients on new dataset
  - predictive ability on new dataset
    - one model can be better at estimation, but another better at prediction

Example: surgical unit

Example: surgical unit

- Random sample of 54 patients undergoing a liver operation
- Response `surv` or `lsurv` post-operation survival (or log-survival) time
- Predictor variables
    - `blood` blood clotting score
    - `prog` prognostic index
    - `enz` enzyme function score
    - `liver` liver function score
    - `age` in years
    - `female` gender, 0=male, 1=female
    - `modAlc` and `heavyAlc` alcool use

Getting to know the data

```
> X <- read.table('/Users/ovitek/Dropbox/Olga/Teaching/CS6220/Fall15/Le
> dimnames(X)[[2]] <- c('blood', 'prog', 'enz', 'liver',
+  'age', 'female', 'modAlc', 'heavyAlc', 'surv', 'lsurv')
> dim(X)
```

```
[1] 54 10
```

```
> head(X)
```

```
  blood prog enz liver age female modAlc heavyAlc surv lsurv
1   6.7   62  81  2.59  50      0      1        0  695 6.544
2   5.1   59  66  1.70  39      0      0        0  403 5.999
3   7.4   57  83  2.16  55      0      0        0  710 6.565
4   6.5   73  41  2.01  48      0      0        0  349 5.854
5   7.8   65 115  4.30  45      0      0        1 2343 7.759
6   5.8   38  72  1.42  65      1      1        0  348 5.852
```

```
> sum(is.na(X))
```

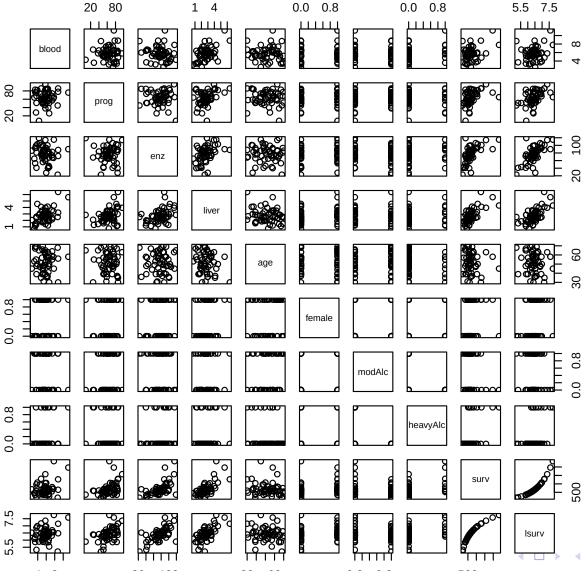
```
[1] 0
```

Getting to know the data

```
> round(cor(X[,-c(9:10)]), digits=2)

         blood  prog   enz liver   age female modAlc heavyAlc
blood     1.00  0.09 -0.15  0.50 -0.02   0.04  -0.10     0.22
prog      0.09  1.00 -0.02  0.37 -0.05   0.12   0.13    -0.08
enz      -0.15 -0.02  1.00  0.42 -0.01   0.14  -0.09     0.12
liver     0.50  0.37  0.42  1.00 -0.21   0.30  -0.02     0.13
age      -0.02 -0.05 -0.01 -0.21  1.00   0.01   0.15    -0.11
female    0.04  0.12  0.14  0.30  0.01   1.00   0.04    -0.06
modAlc   -0.10  0.13 -0.09 -0.02  0.15   0.04   1.00    -0.51
heavyAlc  0.22 -0.08  0.12  0.13 -0.11  -0.06  -0.51     1.00
```

Getting to know the data

```
> pairs(X)
```



Exhaustive subset selection

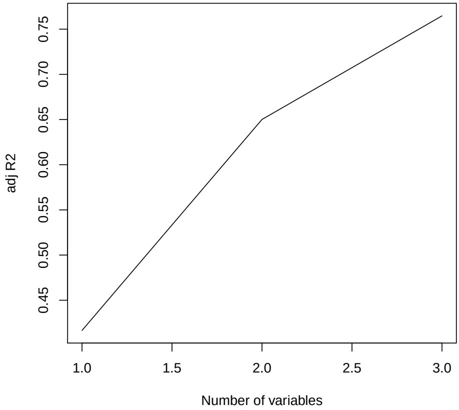
```
> library(leaps)
> # By default - exhaustive search
> regfit.full <- regsubsets(lsurv ~ ., nvmax=3, data=X[,-9])
> reg.summary <- summary(regfit.full)
> names(reg.summary)

[1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj

> reg.summary$which

  (Intercept) blood  prog  enz liver   age female modAlc heavyAlc
1        TRUE FALSE FALSE TRUE FALSE FALSE  FALSE  FALSE    FALSE
2        TRUE FALSE  TRUE TRUE FALSE FALSE  FALSE  FALSE    FALSE
3        TRUE FALSE  TRUE TRUE FALSE FALSE  FALSE  FALSE     TRUE
```

Exhaustive subset selection

```
> reg.summary$rsq

[1] 0.4275662 0.6632899 0.7780337

> reg.summary$adjr2

[1] 0.4165579 0.6500855 0.7647157

> which.max(reg.summary$adjr2)

[1] 3

> coef(regfit.full, 3)

(Intercept)        prog         enz    heavyAlc
 4.29068119  0.01493053  0.01447422  0.42907938
```

Exhaustive subset selection

```
> plot(reg.summary$adjr2, xlab='Number of variables',
+      ylab='adj R2', type='l')
```



Larger number of predictors: heuristics

- Forward selection
  - start with no variables
  - add one variable with best F-value (only if p-value < `sle`)
  - add the next variable with best F-value given the previous variables in the model (only if p-value < `sle`)
  - stop if no variables can be added with p-value < `sle`
- Backward elimination
  - start with all the variables
  - delete the variable that has the smallest extra SS (only if p-value > `sls`)
  - delete the next variable that has the smallest extra SS (only if p-value > `sls`)
  - stop when all variables have p-value < `sls`

Larger number of predictors: heuristics

- Stepwise search
    - start with no variables
    - add variables sequentially as in forward selection, using `sle`
    - once a variable is added, remove all insignificant variables as in backward elimination, using `sls`
    - stop when nothing can be added, and nothing non-significant can be removed
    - fix `sle` $\leq$ `sls` to void cycling.

Example: forward selection

```
> regfit.full1 <- regsubsets(lsurv ~ ., method='forward',
+  data=X[,-9])
> reg.summary1 <- summary(regfit.full1)
> #reg.summary1
> reg.summary1$adjr2

[1] 0.4165579 0.6500855 0.7647157 0.8159970 0.8205081 0.8234494 0.822597
[8] 0.8187737

> which.max(reg.summary1$adjr2)

[1] 6
```

Example: surgical unit

Data-rich situation: independent validation

- Gold standard of validation
- If the number of observations is large, randomly partition the dataset into three parts

1. Training set
    - predictive ability of any model is too optimistic (model fit caters to the training set)
2. Independent variable selection set
    - select predictors that minimize predictive error on this independent set
    - predictive ability of the ”best” model is still too optimistic (variable selection caters to the variable selection set)

Data-rich situation: independent validation

3 Independent validation set

- verifies the predictive ability of the model based on these completely independent data

$$MSPR = \frac{\sum_{i=1}^{n^*} (Y_i - \hat{Y}_i)^2}{n^*}$$

- $n^* = \#$ of observations in validation set

Data-poor situation: cross-validation

- If # of observations is relatively small, but larger than # of variables, randomly partition the dataset into three parts
  - (1) training, (2) var. selection, (3) validation
- Iteratively use each part for training / variable selection / validation
  - each observation will play each role once
  - a value of predictive error for each observation
  - better use of the resources
  - may have a different model at different iteration of cross-validation
- See JWHT Sec. 6.5.3 for R code
  - Or, use `library(DAAG)`
    Maindonald, J.H. and Braun, W.J. (3rd Ed., 2010) *Data Analysis and Graphics Using R*

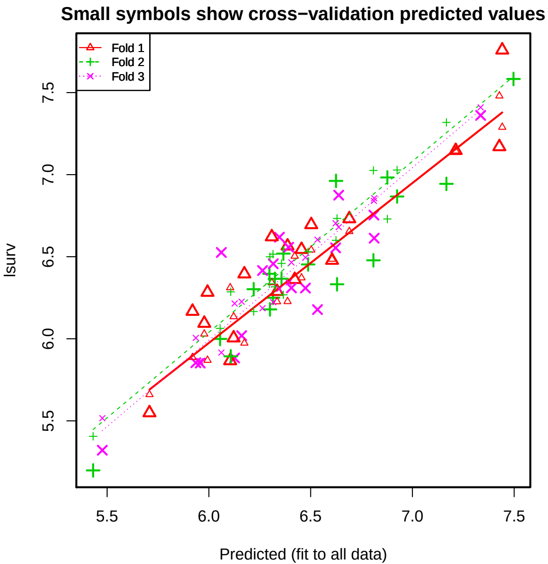
Example: cross-validation [Long output. Run in R]

```
> library(DAAG)
> lm.full <- lm(lsurv ~ ., data=X[,-9])
> CVlm(X[,-9], lm.full)
```

```
Analysis of Variance Table

Response: lsurv
          Df Sum Sq Mean Sq F value  Pr(>F)
blood      1   0.78    0.78   17.73 0.00012 ***
prog       1   2.59    2.59   59.11 9.8e-10 ***
enz        1   6.33    6.33  144.63 1.2e-15 ***
liver      1   0.02    0.02    0.56 0.45767
age        1   0.13    0.13    2.89 0.09615 .
female     1   0.05    0.05    1.19 0.28067
modAlc     1   0.09    0.09    2.03 0.16137
heavyAlc   1   0.85    0.85   19.31 6.7e-05 ***
Residuals 45   1.97    0.04
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.


fold 1
```

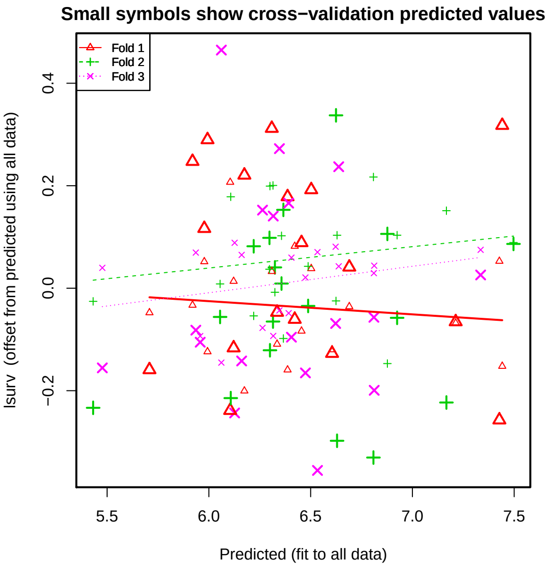
Visualization of cross-validation: fit

```
> CVlm(X[,-9], lm.full, printit=FALSE, plotit='Observed')
```



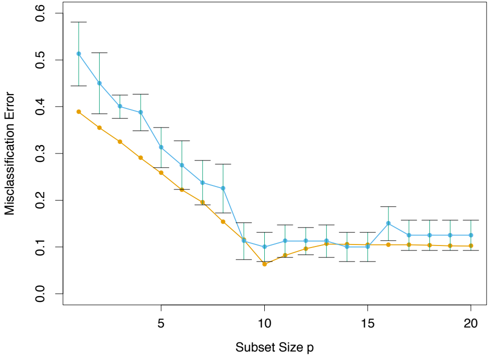
Visualization of cross-validation: residuals

```
> CVlm(X[,-9], lm.full, printit=FALSE, plotit='Residual')
```



Cross-validation and variable selection

- Orange line: in-sample prediction error
- Blue line: cross-validated prediction error
  - Error bars are obtained over each fold (alternatively, by repeatedly partitioning data into folds)



From Hastie, Tibshirani, Friedman *The elements of Statistical Learning*, 2nd Ed., Springer