Practical 2 Prediction accuracy and pitfalls

Huanwei Wang (huanwei.wang@uq.edu.au)

2022-06-23

We will learn how to evaluate the prediction accuracy and demonstrate some pitfalls for genetic prediction.

Further information: https://cnsgenomics.com/data/teaching/SISG/module_10/Mod10_Session7Naomi/Practical7/

Prediction accurarcy

R-square for quantitative traits

```
### set the working directory
#setwd("~/module4/prac2/")

### read phenotype value and prediction scores
pheno = read.table("quan.pheno", header=T)
prs = read.table("quan.prs", header=T)
covar = read.table("quan.covar", header=T)
data = merge(pheno, prs, by="ID")
data = merge(data, covar, by="ID")

### linear regression
lmR = lm(pheno ~ age + sex, data=data)  ### reduced module
lmF = lm(pheno ~ age + sex + prs, data=data) ### full module
```

```
### look at the summary results for the two models
summary(lmR)
summary(lmF)
```

```
### incremental r-square
summary(lmF)$"r.square" - summary(lmR)$"r.square"
```

```
## [1] 0.09699574
```

Nagelkerke's R-sqaure

```
### read phenotype value and prediction scores
pheno = read.table("bin.pheno", header=T)
prs = read.table("bin.prs", header=T)
covar = read.table("bin.covar", header=T)
data = merge(pheno, prs, by="ID")
data = merge(data, covar, by="ID")

### logistic regression
glmR = glm(pheno ~ age + sex, data=data, family=binomial(logit))  ### reduced module
glmF = glm(pheno ~ age + sex + prs, data=data, family=binomial(logit))  ### full module
```

```
### look at the summary results for the two models
summary(glmR)
summary(glmF)
```

```
### log-likelihood
N = nrow(data)
LLF = logLik(glmF)
LLR = logLik(glmR)

### Cox&Snell R2
CSv <-  1-exp((2/N)*(LLR[1]-LLF[1]))
CSv
```

```
## [1] 0.06630625
```

```
### Nagelkerke's R2
NKv <- CSv/(1-exp((2/N)*LLR[1]))
NKv
```

```
## [1] 0.08864474
```

AUC

```
### AUC
### install.packages('pROC')
library('pROC')
aucF = auc(data$pheno, glmF$linear.predictors)  ### AUC for reduced module
aucR = auc(data$pheno, glmR$linear.predictors)  ### AUC for full module
aucF; aucR
```

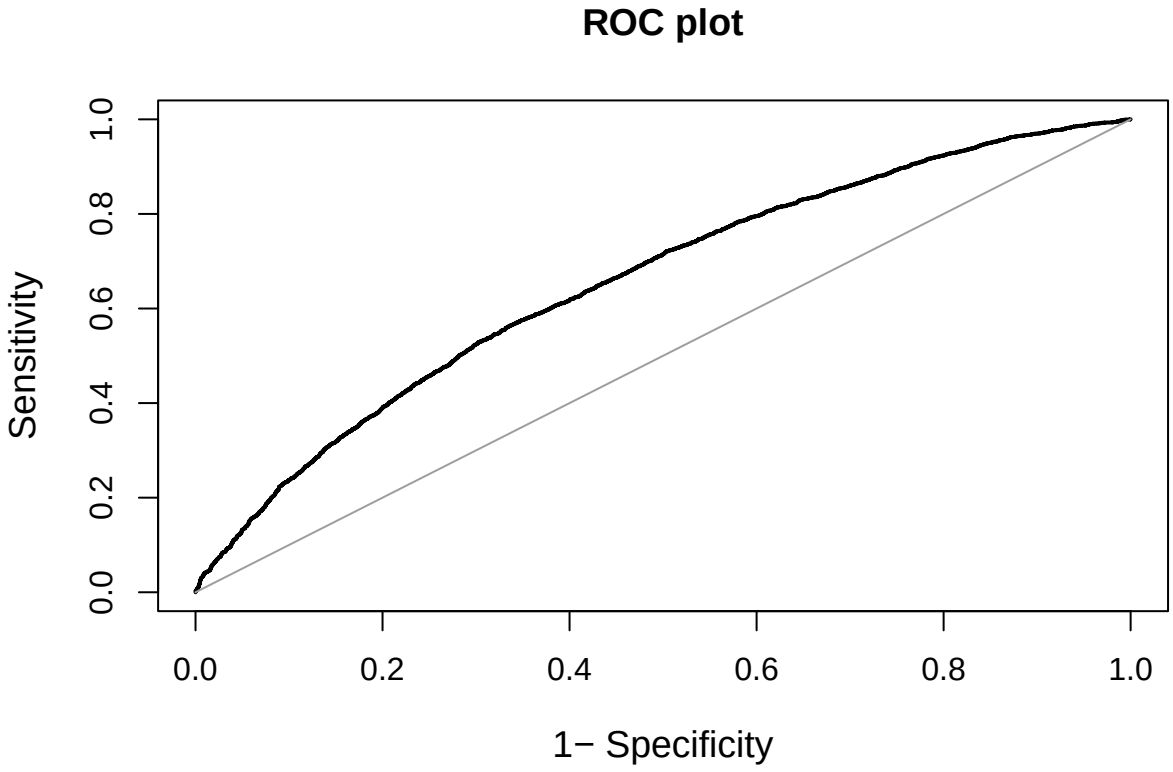
```
## Area under the curve: 0.6553
```

```
## Area under the curve: 0.5508
```

```
aucF - aucR
```

```
## [1] 0.1044923
```

```
### draw the ROC
#install.packages("PredictABEL")
library(PredictABEL)
plotROC(data=data, cOutcome=2, predrisk=glmF$linear.predictors)
```




```
## AUC [95% CI] for the model 1 :  0.655 [ 0.645  -  0.666 ]
```

Decile Odds Ratio

```
### cut into deciles
data$decile = cut(data$prs, breaks=c(quantile(data$prs, probs=seq(0,1,by=0.1))), labels=1:10, include.l
```

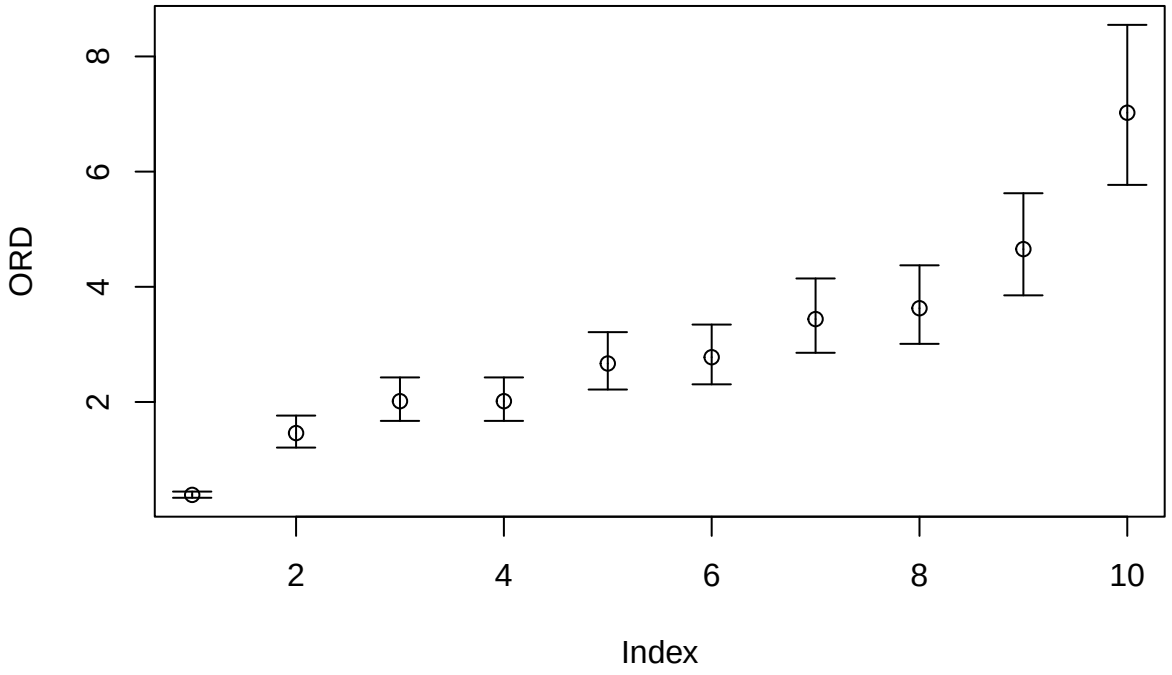
```
### calculate manually the odds in each decile
#### install.packages("tidyverse")
library(dplyr)
data %>% group_by(decile) %>%
    summarise(n_case=sum(pheno==1), n_control=sum(pheno==0)) %>%
    mutate(odds = n_case/n_control) %>%
    mutate(ORs = odds/odds[1])
```

```
## # A tibble: 10 x 5
##    decile n_case n_control  odds   ORs
##    <fct>   <int>     <int> <dbl> <dbl>
##  1 1         279       721 0.387  1
##  2 2         361       639 0.565  1.46
##  3 3         438       562 0.779  2.01
##  4 4         438       562 0.779  2.01
##  5 5         508       492 1.03   2.67
##  6 6         518       482 1.07   2.78
##  7 7         571       429 1.33   3.44
##  8 8         584       416 1.40   3.63
##  9 9         643       357 1.80   4.65
## 10 10        731       269 2.72   7.02
```

```r
### calculate ORs using logistic regression
glmD <- glm(pheno ~ decile, data = data, family = binomial(logit))
### Odds for being a case compared to control in each decile
ORD <- exp(glmD$coefficients)
ORD
```

```
## (Intercept)     decile2     decile3     decile4     decile5     decile6
##   0.3869626   1.4599481   2.0140435   2.0140435   2.6682694   2.7772424
##     decile7     decile8     decile9    decile10
##   3.4396153   3.6278605   4.6545084   7.0225713
```

```r
### Plot odds
ORDL <- exp(glmD$coefficients-1.96*summary(glmD)$coefficients[,2])
ORDH <- exp(glmD$coefficients+1.96*summary(glmD)$coefficients[,2])
plot(ORD,ylim=c(min(ORDL),max(ORDH)))
arrows(seq(1,10,1), ORD, seq(1,10,1), ORDH, angle=90,length=0.10) # Draw error bars
arrows(seq(1,10,1), ORD, seq(1,10,1), ORDL, angle=90,length=0.10) # Draw error bars
```



Variance explained on liability scale

```r
### function to convert R-square from 0-1 observed scale
### to liability scale
h2l_R2 <- function(k, r2, p) {
  # k baseline disease risk
  # r2 from a linear regression model of genomic profile risk score
  # p proportion of sample that are cases
  # calculates proportion of variance explained on the liability scale
  # from ABC at http://www.complextraitgenomics.com/software/
  # Lee SH, Goddard ME, Wray NR, Visscher PM. (2012) A better coefficient of determination for genetic
  x = qnorm(1-k)
  z = dnorm(x)
  i = z/k
  C = k*(1-k)*k*(1-k)/(z^2*p*(1-p))
```

```
  theta = i*((p-k)/(1-k))*(i*((p-k)/(1-k))-x)
  h2l_R2 = C*r2 / (1 + C*theta*r2)
  return(h2l_R2)
}
K=0.1  ## population prevalence
P = sum(data$pheno)/nrow(data)  ### propotion of cases in the sample
P
```

```
## [1] 0.5071
```

```
### linear regression with 0/1 values
lmR = lm(pheno~age+sex, data=data)
lmF = lm(pheno~age+sex+prs, data=data)
R2v = summary(lmF)$"r.square" - summary(lmR)$"r.square"
R2v
```

```
## [1] 0.06580439
```

```
### convert to liability scale
h2l_R2(K,R2v,P)
```

```
## [1] 0.07114465
```

Pitfalls

Simulate data

```
### we simulate data now
set.seed(612)    ### set a seed for reproducation
n=1000; m=100    ### n: sample size; m: the number of SNPs

### simulate gneotype from binomial distribution
### minor allele frequency from uniform distribution
mafs = runif(m, 0.05, 0.5)
x = do.call("cbind", lapply(1:m, function(x) rbinom(n, 2, mafs[x])))
colnames(x) = paste("SNP", 1:m, sep="")

### simulate a phenotype from an independent standard normal distribution
### null hypothesis: prediction accuracy should be 0
y = rnorm(n)
data = data.frame(y, x)
```

Directly report R2 in the discovery sample

```
### E(R2) = m/n when m<n
summary(lm(as.formula(paste("y~", paste("SNP", 1:m, sep="", collapse="+"), sep="")),data=data.frame(data
```

```
## [1] 0.08951179
```

Winner's curse

```
### perform GWAS: association using R lm() function
### instead of Plink software (only practical for small data)
### select the top 10 SNPs
pvals = sapply(1:m, function(i) coef(summary(lm(y~x[,i])))[2,4])
```

```
### No p-values passed Bonferroni threshold of 5e-4 (0.05/100)
head(sort(pvals))
```

```
## [1] 0.0007738269 0.0144049898 0.0185012440 0.0612186306 0.0612745215
## [6] 0.0616904469
```

```
top10SNPs = colnames(x)[head(order(pvals), 10)]
summary(lm(as.formula(paste("y~", paste(top10SNPs, collapse="+"), sep="")),data=data))$"r.square"
```

```
## [1] 0.04027331
```

```
### 0.04027/0.0895(=0.4499) >> 10/100(=0.1)
```

Estimate the SNPs in the total sample, and evaluate in the target sample

```
### first 900 samples into discovery set
y_dis = y[1:900]
x_dis = x[1:900,]
### the remaining 100 samples into target set
y_target = y[901:1000]
x_target = x[901:1000,]

### estimate effect sizes in the total sample
b_total = sapply(1:m, function(i) coef(summary(lm(y~x[,i])))[2,1])
### evaluate the PRS in the target sample
prs1 = x_target %*% b_total
summary(lm(y_target~prs1))$"r.square"
```

```
## [1] 0.1573479
```

```
### estimate effect sizes only in the discovery sample
b_dis = sapply(1:m, function(i) coef(summary(lm(y_dis~x_dis[,i])))[2,1])
### evaluate the PRS in the target sample
prs2 = x_target %*% b_dis
summary(lm(y_target~prs2))$"r.square"
```

```
## [1] 0.001510061
```