Module 2: Understanding Ridge-Regression in Genomic Predictions

Fundamentals of Genomic Prediction and Data-Drive Crop Breeding

(August 4-8, 2025)

Waseem Hussain
Senior Scientist-I
International Rice Research Institute
Rice Breeding Innovations Platfrom
waseem.hussain@cgiar.org
whussain2.github.io

Mahender Anumalla
Scientist-I
International Rice Research Institute
South-Asian Hub, Hyderabad
m.anumalla@cgiar.org

August 3, 2025

Contents

Introduction

Purpose of this session is how **Ridge Regression** overcomes limitations of **Ordinary Least Squares (OLS)** and also demonstrate the **n«<p** problem in genomic predictions, called as **Curse on Dimensionality**. Here *n* number of genotypes and *p* are predictors that is number of markers.

Read these resources for more details **Resource 1**; **Resource 2** and **Resource 3**.

Shrinkage Explaination: How it Works

Here we will see a quick example how λ is avoiding the problems of OLS

Creat a Hypothetical Matrix

```
> # Matrix, 3 rows and 5 columns
>   set.seed(1)
>   n <- 3
>   m <- 5
>   X <- matrix(rbinom(n = n * m, size = 2, prob = 0.5), nrow = n, ncol = m)
>   X
```

```
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    2    2    0    1
[2,]    1    0    1    0    1
[3,]    1    2    1    0    2
```

Now Get Determinent

More on determinant of matrix, **click here**

```
>   det(t(X) %*% X)
```

```
[1] 0
```

Role of Lambda as Shrinkage Factor

```
> # determinant
>   det(t(X) %*% X + diag(1, m))
```

```
[1] 96
```

Please note now how determinant is obtained as compared to zero without λ

Example with Real Data

Here we will be using rice SNP marker data available at http://ricediversity.org/data/index.cfm. The rice data is already downloaded in the folder. The marker data includes 44,100 SNP markers for 413 diverse accessions/genotypes of O. sativa.

The marker data is on .ped format and will will convert it into numeric format (0,1,2). For more on format conversion on marker dates check our resource on [**GitHub Page. For this we will use BGLR R package.

Load Genotype Data

```
> # Load Package
>   library(BGLR)
>   rm(list=ls()) # Remove previous history
> # Read marker file in .ped format
>   Geno<-read_ped("./Data/sativas413.ped")
> # Set dimenions
>   p=Geno$p
>   n=Geno$n
>   Geno=Geno$x
> # Now load .fam file having genotype/acession names
>   FAM <- read.table("./Data/sativas413.fam")
> # Now read .mpa file containing map information
>   MAP <- read.table("./Data/sativas413.map")
> # Let us now recode the marker data in .ped file
>   Geno[Geno == 2] <- NA  # Converting missing data to NA
>   #Geno[Geno == 0] <- 0   # Converting 0 data to 0
>   #Geno[Geno == 1] <- 1   # Converting 1 to 1
>   Geno[Geno == 3] <- 2   # Converting 3 to 2
> # Now convert the marker data into matrix and transponse and check dimensions
>   Geno <- matrix(Geno, nrow=p, ncol=n, byrow=TRUE)
>   Geno <- t(Geno)
>   dim(Geno)
```

```
[1]   413 36901
```

```
>   Geno[1:10, 1:10]
```

```
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
 [1,]    0    0    0    0    0    0    0    0    0     0
 [2,]    2    2    0    2    2   NA    0    2    2     2
 [3,]    2    2    0    2    2    2    0    2    2     2
 [4,]    2    2    2    0    2    0    2    2    2     0
 [5,]    2    2    0    2    2    2    0    2    2     2
 [6,]    0    0    0    0    0    0    0    0    0     0
 [7,]    0    0    0    0    0    0    0    0    0     0
```

```
 [8,]    0    0    0    0    0    0    0    0    0     0
 [9,]    0    0    0    0    0    0    0    0    0     0
[10,]    0    0    0    0    0    0    0    0    0     0
```

```
> # Convert the missing, impute as mean
>   for (j in 1:ncol(Geno)) {
+     Geno[, j] <- ifelse(is.na(Geno[, j]), mean(Geno[, j], na.rm = TRUE),
+                         Geno[,j])
+     }
>   Geno[1:5, 1:4]
```

```
     [,1] [,2] [,3] [,4]
[1,]    0    0    0    0
[2,]    2    2    0    2
[3,]    2    2    0    2
[4,]    2    2    2    0
[5,]    2    2    0    2
```

```
> # Now assign the row and column names to marker file
>   colnames(Geno)<-MAP$V2
> # Adding line names stored in column second and pasted NSFTV_ID_ to each line name.
>   row.names(Geno)<-paste0("NSFTV_",FAM$V2)
>   #saveRDS(Geno,"./Data/Geno.coverted.rds")
```

Note: You can read converted numeric formatted marker data file named as ***Geno.coverted.rds** directly from folder and skip the above step.

Load the Phenotype Data

The phenotypic data includes 34 traits phenotyped for 413 acessions. We will use just one trait for OLS estimations.

```
> # Phenotypic data
>   pheno<-read.csv(file="./Data/rice.csv",
+                   header=TRUE)
> # Convert the missing data into mean
>   for (j in 1:ncol(pheno)) {
+     pheno[, j] <- ifelse(is.na(pheno[, j]), mean(pheno[, j], na.rm = TRUE),
+                          pheno[, j])
+   }
```

Now Fit OLS

Here we will perform simple marker regression through OLS using the equation:

$$\beta = (X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}Y$$

where,

X : is matrix of fixed effects
Y : is response variableβ : are SNP effects

We will use X with only 20 Markers n»p.

We will use first variable, Flowering at arkansas as phenotype trait.

```
> # Create a phenotype vector
>   pheno1 <- as.vector(pheno$Flowering.time.at.Arkansas) # phenotype vector
> # Create intercept
>   intercept <- rep(1, length(pheno1)) # intercept vector
> # Create an X matrix of SNPs (first 20) and add intercept
>   X <- cbind(intercept, Geno[,1:20]) # the intercept and the SNP matrix including the first 100 SNPs
> # Check dimesnions
>   length(pheno1)
```

```
[1] 413
```

```
>   dim(X)
```

```
[1] 413  21
```

```
> # Fit OLS through equation
>   ols1<- solve(t(X) %*% X) %*% t(X) %*% pheno1
>   head(ols1) # estimates of the intercept and the first five SNP effects
```

```
               [,1]
intercept  84.655090
id1000001   1.090449
id1000003  11.707490
id1000005   6.618623
id1000007 -13.843865
id1000008  -7.603096
```

Fitting Markers p » n

Check the error, matrix is singular because p»>n

```
> # Create subset of markers 1000
>   X2 <- cbind(intercept, Geno[, 1:1000]) # the intercept and the whole SNP matrix
>   dim(X2)
```

```
[1]  413 1001
```

```
> # Fit all Markers
>   # use the solve() function
>   #ols2<- solve(t(X2) %*% X2) %*% t(X2) %*% pheno1
>   # Check the error
>   # use the lm() function
>   summary(lm(pheno1 ~ -1 + X2)) # check the warning
```

```
Call:
lm(formula = pheno1 ~ -1 + X2)

Residuals:
ALL 413 residuals are 0: no residual degrees of freedom!

Coefficients: (588 not defined because of singularities)
              Estimate Std. Error t value Pr(>|t|)
X2intercept  9.977e+01        NaN     NaN      NaN
X2id1000001 -1.411e+15        NaN     NaN      NaN
X2id1000003  1.693e+15        NaN     NaN      NaN
```

```
X2id1000005  1.752e+14        NaN     NaN      NaN
X2id1000007  3.434e+14        NaN     NaN      NaN
X2id1000008 -1.066e+14        NaN     NaN      NaN
X2id1000011  3.638e+04        NaN     NaN      NaN
X2id1000013  5.063e+13        NaN     NaN      NaN
X2id1000015 -1.848e+14        NaN     NaN      NaN
X2id1000016  6.173e+13        NaN     NaN      NaN
X2id1000020 -7.184e+13        NaN     NaN      NaN
X2id1000024  6.399e+02        NaN     NaN      NaN
X2id1000026 -3.240e+14        NaN     NaN      NaN
X2id1000027  1.306e+01        NaN     NaN      NaN
X2id1000030 -1.050e+04        NaN     NaN      NaN
X2id1000043 -1.396e+04        NaN     NaN      NaN
X2id1000051  3.240e+14        NaN     NaN      NaN
X2id1000057  3.503e+05        NaN     NaN      NaN
X2id1000058 -6.258e+13        NaN     NaN      NaN
X2id1000062 -1.978e+14        NaN     NaN      NaN
X2id1000074 -2.001e+14        NaN     NaN      NaN
X2id1000075  9.560e+13        NaN     NaN      NaN
X2id1000079 -8.330e+13        NaN     NaN      NaN
X2id1000080 -6.197e+13        NaN     NaN      NaN
X2id1000086 -3.025e+04        NaN     NaN      NaN
 [ reached getOption("max.print") -- omitted 976 rows ]

Residual standard error: NaN on 0 degrees of freedom
Multiple R-squared:      1, Adjusted R-squared:    NaN
F-statistic:   NaN on 413 and 0 DF,  p-value: NA
```

Now Add Shrinkage Factor

Here we will add $\lambda = 1$ and see how we get it estimates.

```
> # Same 1000 markers
>   X2 <- cbind(intercept, Geno[, 1:1000]) # the intercept and the whole SNP matrix
>   dim(X2)
```

```
[1]  413 1001
```

```
>   # use the solve() function
>   ols2<- solve(t(X2) %*% X2+diag(1, 1001)) %*% t(X2) %*% pheno1
> # Check the error
```

No error of singularity, means matrix is inverted and we get estimates. This is one example of moving from basic linear regression models to Ridge Regression which is all about adding penalty factor to avoid **Singularity**

Assignment: Why n«p is not a problem in GWAS or QTL Mapping?

For any suggestions or comments, please feel to reach at waseem.hussain@irri.org; and m.anumalla@irri.org ***