ASMNW - Übung 4

Peter von Rohr

2018-03-24

Aufgabe 1: Einfaches Beispiel mit nur einem SNP

Wir betrachten ein einfaches Beispiel mit nur 10 Tieren, welche für einen einzigen SNP typisiert sind. Die folgende Tabelle gibt eine Übersicht über die Daten mit den SNP-Allelen und den phänotypischen Beobachtungen.

Animal	Phentype	SnpAllele1	SnpAllele2
1	2.03	1	1
2	3.54	1	2
3	3.83	1	2
4	4.87	2	2
5	3.41	1	2
6	2.34	1	1
7	2.65	1	1
8	3.76	1	2
9	3.69	1	2
10	3.69	1	2

Wir nehmen an die Tiere seien nicht verwandt miteinander. Somit können wir die Beziehung zwischen dem einen SNP und den phänotypischen Beobachtungen mit einem einfachen Regressionsmodell testen. Unser Modell lautet:

$$y = 1_n\mu + Wg + e \tag{1}$$

wobei

- y Vektor der Länge n mit phänotypischen Beobachtungen
- μ allemeines Mittel, welches fixe Effekte repräsentiert
- 1_n Vektor der Länge n mit lauter Einsen
- g additiver Effekt des Marker-SNP
- W Inzidenzmatrix, welche die Beobachtungen zum Marker-Effekt verbindet
- e Vektor der zufälligen Resteffekte

Die Inzidenzmatrix W hat n Zeilen und so viele Kolonnen, wie SNP-Marker. Für unser Beispiel hat die Matrix W somit 1 Kolonne. Die Elemente der Matrix W zählen die Anzahl Allele mit positiver Wirkung. In diesem Beispiel sei das Allel "2".

Ihre Aufgabe

Stellen Sie das Modell aus Gleichung (1) für den gegebenen Datensatz auf und bestimmen Sie welche Modellkomponenten bekannt und welche unbekannt sind.

Aufgabe 2: Least Squares Lösungen

Da wir 10 Beobachtungen und nur einen SNP betrachten ist die Bedingung für die Schätzung der unbekannten Parameter mit Least Squares erfüllt. Somit können wir die unbekannten Parameter μ und g mit der folgenden Gleichung schätzen.

$$\begin{bmatrix} \hat{\mu} \\ \hat{g} \end{bmatrix} = \begin{bmatrix} 1_n^T 1_n & 1_n^T W \\ W^T 1_n & W^T W \end{bmatrix}^{-1} \begin{bmatrix} 1_n^T y \\ W^T y \end{bmatrix} \tag{2}$$

Berechnen Sie aufgrund der Gleichungen in (2) die Lösungen für $\hat{\mu}$ und $\hat{g}$.

Aufgabe 3

Überprüfen Sie die unter Aufgabe 2 erhaltenen Least Squares Lösungen für μ und g mit der Funktion `lm()` R.

Hinweise

- Lesen Sie die Daten aus der in Aufgabe 1 gezeigten Tabelle in den Dataframe namens `dfAufgabe3` ein.

```
nAnzTiere <- 10
dfAufgabe3 <- data.frame(
  Animal = c(1:nAnzTiere),
  Phentype = c(2.03, 3.54,3.83,4.87,3.41,2.34,2.65,3.76,3.69,3.69),
  SnpAllele1 = c(1,1,1,2,1,1,1,1,1,1),
  SnpAllele2 = c(1,2,2,2,2,1,1,2,2,2))
```

- Fügen Sie dem Dataframe eine zusätzliche Kolonne namens `Genotype`hinzu, welche die Genotypen-Codes enthält. Diese Codes entsprechen den Anzahl an "2" Allelen mit positiver Wirkung.
- Verwenden Sie die phänotypischen Beobachtungen in `dfAufgabe3$Phenotype` als Zielgrösse und `dfAufgabe3$Genotype` als erklärende Variable und passen Sie das Regressionsmodell mit der Funktion `lm()` an.

Aufgabe 4

Gegeben ist ein Datensatz in der Datei asmas_w05_u04_lasso.txt, welcher Genotypen an 100 SNP-Loci und Beobachtungen für ein bestimmtes Merkmal für insgesamt 50 Tiere enthält. Sie können den gesamten Datensatz mit dem folgenden Statement in eine Matrix einlesen.

```
## mat_lasso_data <- matrix(scan("asmas_w05_u04_lasso.txt"), nrow = 50, byrow = TRUE)
```

Betrachten wir uns die ersten fünf Zeilen und die ersten fünf Kolonnen dieser Matrix sehen diese wie folgt aus.

```
##           [,1] [,2] [,3] [,4] [,5]
## [1,] -40.39872   -1    0   -1   -1
## [2,] -46.35871   -1   -1   -1    0
## [3,] -33.60278   -1   -1   -1   -1
## [4,] -48.47177    0   -1   -1   -1
## [5,] -38.82089   -1   -1    0   -1
```

Daraus wird ersichtlich, dass die Beobachtungen aller Tiere in der ersten Kolonne der Datenmatrix `mat_lasso_data` sind und die Genotypen in den Kolonnen 2 bis 101 . Für die Anpassung eines linearen Modells mit LASSO verwenden wir die Funktion `glmnet()` aus dem gleichnamigen Package `glmnet`. Wir verwenden jetzt also alle SNP-Genotypen als erklärende Variablen und die Beobachtungswerte sind unsere Zielgrössen.

Ihre Aufgaben

- Verwenden Sie das folgende R-Statement für die Schätzung der SNP-Effekte mit LASSO

```
require(glmnet)
fitsnp <- glmnet(x = mat_lasso_data[, -1], y = mat_lasso_data[, 1])
```

- Visualisieren Sie die Abhängigkeit zwischen dem Wert von λ und der Anzahl von erklärenden Variablen, welche nicht 0 sind.

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

- Machen Sie eine Kreuzvalidierung um den Wert vom λ zu bestimmen

```
cvfitsnp <- cv.glmnet(x = mat_snp, y = vec_y)
```

- Stellen Sie die Resultate der Kreuzvalidierung mit der Funktion `plot()` dar.

```
plot(cvfitsnp)
```

- Im Plot der Kreuzvalidierungsresultate gibt es zwei gestrichelte Linien, welche zwei spezielle λ-Werte markieren. Der erste Wert ist das Minimum aller λ-Werte und der zweite ist der Wert, welcher die meisten Variablen auf 0 setzt aber nicht weiter als eine Standardabweichung vom minimalen Wert des mittleren quadrierten Fehler entfernt ist. Die beiden λ-Wert erhalten Sie mit

```
cvfitsnp$lambda.min
cvfitsnp$lambda.1se
```

- Finden Sie alle Koeffizienten, welche nicht 0 sind, für die beiden λ-Werte und vergleichen Sie diese mit den wahren Werten aus der Simulation

```
coefmin <- coef(cvfitsnp, s = "lambda.min")
(cofminnz <- coefmin[coefmin[, 1] != 0,])
```

```
coef1se <- coef(cvfitsnp, s = "lambda.1se")
(coef1senz <- coef1se[coef1se[, 1] != 0, ])
```

Die wahren SNP-Positionen aus der Simulation lauten:

```
(vec_sign_snp_idx <- c(73,54,26,30,7))
```

```
## [1] 73 54 26 30  7
```