BIOST5544_HW3_AWolf

Aaron Wolf

January 25, 2017

```
clin.data= fread('~/Documents/Dropbox/2016:2017/BIOST544/Data/clinical_data.csv')
expr.data = fread('~/Documents/Dropbox/2016:2017/BIOST544/Data/expression_data_probeID.csv')
```

```
##
Read 0.0% of 154 rows
Read 154 rows and 54678 (of 54678) columns from 0.094 GB file in 00:00:04
```

```
annot.data = fread('~/Documents/Dropbox/2016:2017/BIOST544/Data/annotation.csv')
```

Large number of necrotic cells in a tumor can be indicative of a successfuly mounted immune defense Investigate the relationship between gene-expression values in the tumor and the existence and extenet of necrotic tissue. identify genes with expression related to quantity of necrotic tumor tissue (necorsis ~ expression1 + expression2 + expression3) Two potentially useful frameworks are prediction-based (lasso, ridge-regression) and screening-based (permutation testing) Evaluate over-optimism and/or account for multiplicity in your testing

Use prediction-based approach:
My aim here is to find a relationship between the response (percent necrotic cells), and predictors (probe1, probe2, …) by using a linear regression of the form necrotic_cells.pct ~ B1xprobe1 + B2xprobe2 + B3xprobe3 …
Because the number of features is very large, espsecially compared to the number of observations, I wanted to use a feature reduction approach to limit the model complexity and avoid over-fitting. I therefore applied Lasso regularization to limit the number of features selected.
Furthermore, to avoid over-fitting to the data and over-optimism, I split the data into training and test data sets, and when estimating the appropriate lambda for the lasso model, I applied Leave-One-Out-Cross-Validation in the training data.

1. I began by looking for missing data. The clinical data has 2-fewer people than the expression data. I performed an inner-join to create a “sample-keep” list.
2. I split the data into training and validation set. I selected 100 inidivudals randomly from the sample-keep list as a “training” set, and selected the remaining 52 samples as a “test/validation” set.
3. Proceeding only with the training data, I applied lasso feature reduction for a linear regression model in which necrotic_cells.pct is the response variable, and probe expression levels are the predictor variables. (I standardized expression levels before performing modeling). I used Leave-One-Out-Cross-Validation and mean-standard-error to identify the value for lambda that minimized the mse.
4. I then created a predicted response data set using the “test” set from the expression data and the minimum lambda identified through LOOCV. I calculated the MSE for the predicted outcome to the empirical outcome for the “test” set.
5. I retrained the lasso model on the full data set (training and test combined; lasso.full).
6. I selected the coefficients for the lasso.full model using the minimum lambda identified through LOOCV, and identified the probes for all non-zero betas and their associated genes.

```
sample.keep.list <- as.numeric(inner_join(expr.data, clin.data, by=c("centerid","patid"))$V1.x)
```

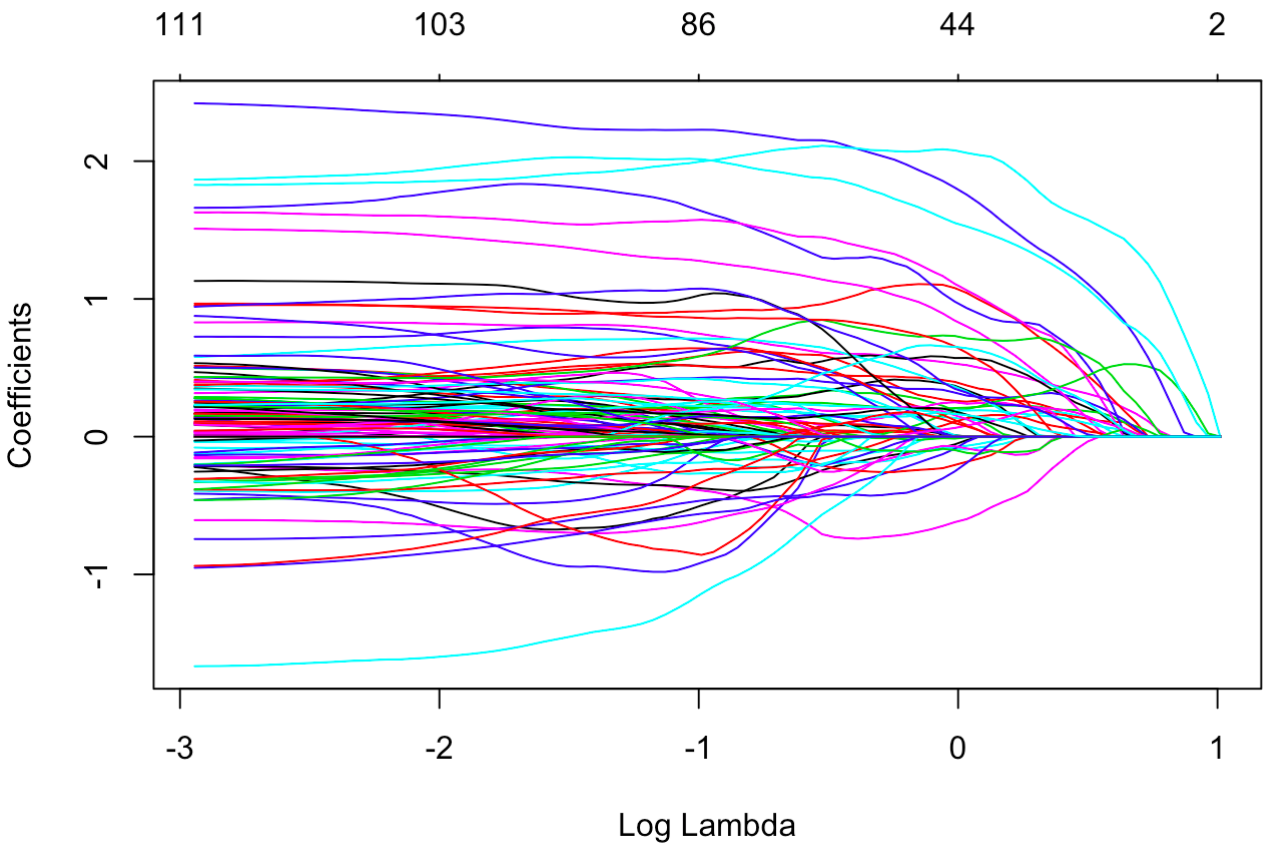
```
set.seed(1)
train = sample(x = sample.keep.list, size = 100, replace = FALSE)

clin.data.train = clin.data[V1 %in% train]
expr.data.train = expr.data[V1 %in% train]

clin.data.test = clin.data[V1 %in% train == FALSE]
test = as.numeric(clin.data.test$V1)

expr.data.test = expr.data[V1 %in% test]
```

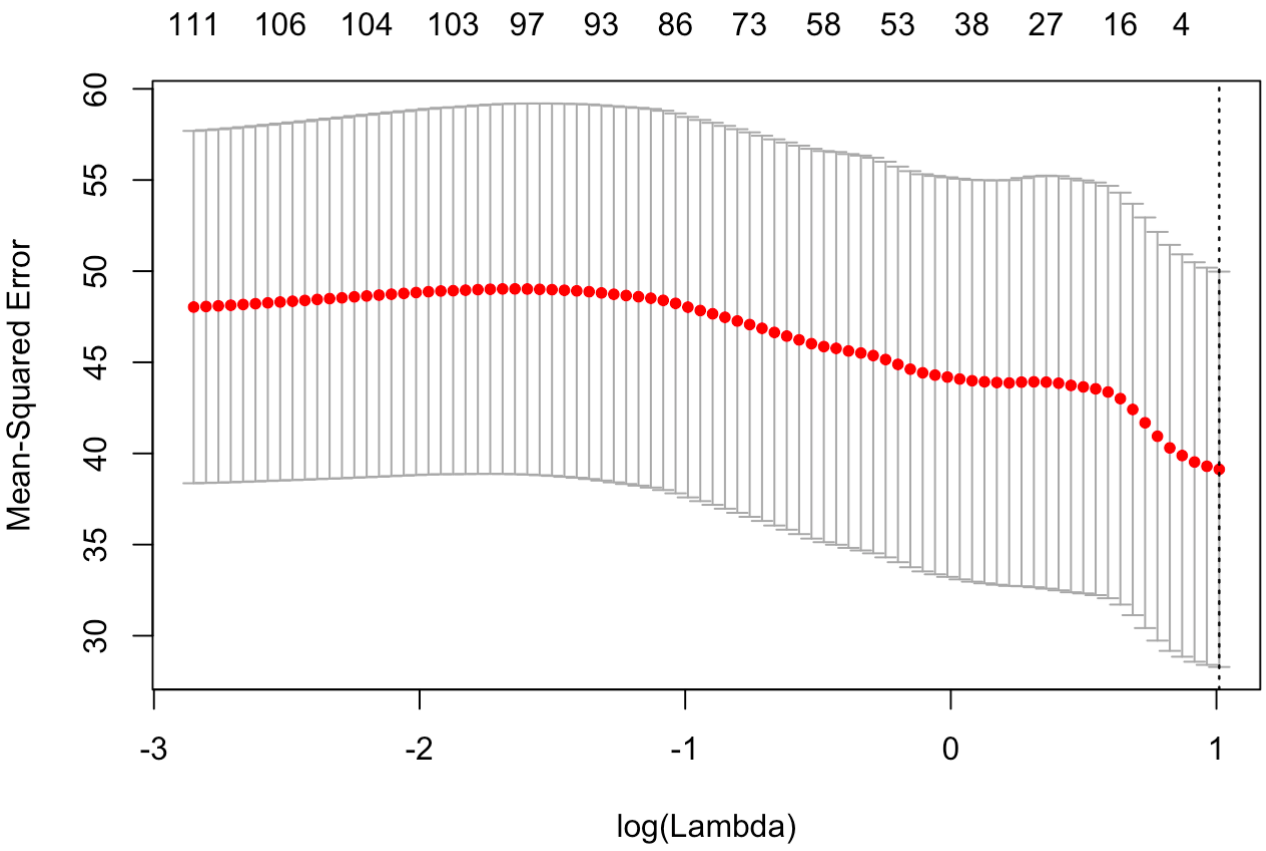
```
lasso.mod = glmnet(x = as.matrix(expr.data.train[,4:ncol(expr.data.train), with=FALSE]), y = as.m
atrix(as.numeric(clin.data.train$necrotic_cells.pct)), alpha = 1, family = 'gaussian', standardiz
e = TRUE)
plot(lasso.mod, xvar="lambda")
```




```
cv.lasso = cv.glmnet(x = as.matrix(expr.data.train[,4:ncol(expr.data.train), with=FALSE]), y = a
s.matrix(as.numeric(clin.data.train$necrotic_cells.pct)), type.measure = "mse", nfolds = 100, alp
ha=1, standardize=TRUE, family="gaussian")
```

```
## Warning: Option grouped=FALSE enforced in cv.glmnet, since < 3
## observations per fold
```

```
plot(cv.lasso)
```




```
min.lambda = round(cv.lasso$lambda.min, digits = 0)

lasso.predict = as.data.table(predict(object = cv.lasso$glmnet.fit, newx = as.matrix(expr.data.te
st[,4:ncol(expr.data.test), with=FALSE])))[,min.lambda,with=FALSE]
MSE = mean((lasso.predict -  as.matrix(as.numeric(clin.data.test$necrotic_cells.pct)))^2)

lasso.full = glmnet(x = as.matrix(expr.data[V1 %in% train | V1 %in% test, 4:ncol(expr.data),
with=FALSE]), y = as.matrix(as.numeric(clin.data$necrotic_cells.pct)), alpha = 1, family = "gauss
ian", standardize = TRUE)
```

```
lasso.coefficients = select(as.data.table(as.matrix(coef(lasso.full)), keep.rownames = TRUE), rn,
 min.lambda+1)
setnames(lasso.coefficients, c('probe','lambda'))
lasso.coefficients.filtered.probes = filter(lasso.coefficients, lambda!=0)$probe

top.probes = filter(lasso.coefficients, lambda!=0)$probe[2:length(lasso.coefficients.filtered.pro
bes)]

associated.genes = annot.data[probset.ids %in% top.probes]

lasso.coefficients.filtered.probes
```

```
## [1] "(Intercept)" "X225115_at"  "X227498_at"
```

```
top.probes
```

```
## [1] "X225115_at" "X227498_at"
```

```
associated.genes
```

```
##    probset.ids gene.names
## 1:  X225115_at      HIPK2
## 2:  X227498_at       SOX6
```