MATH 567: Mathematical Techniques in Data Science
Lab 3

Dominique Guillot

Departments of Mathematical Sciences
University of Delaware

February 22, 2017

Exercise 1

1. Install the package glmnet (if not already installed).
2. Examine the documentation of the glmnet function (?glmnet)
3. Generate random data:

```
n <- 100   # Sample size
p <- 500   # Nb. of variables

true_p <- 10

X <- matrix(rnorm(n*p), nrow=n, ncol=p)

true_beta = matrix(rep(0,p), nrow=p)
true_beta[1:10] = 1

SNR <- 1  # Signal-to-noise ratio
          # = ratio of variances

noise <- matrix(rnorm(n, sd=1/sqrt(SNR)),nrow=n)

y <- X %*% true_beta + noise
```

Note: y depends only on the first 10 predictors.

Exercise 1 (cont.)

4. Fit a ridge regression model to the data (use the options `family="gaussian", alpha=0` in glmnet).
5. What does the `$beta` variable of your ridge model contain? What about `$lambda`?
6. Use the command `matplot` to plot the regression coefficients as a function of λ for the first 10 estimated coefficients. (Note: `matplot` plots the *columns* of a matrix). Use the option type="l".
7. Plot the coefficients $11 : 100$ as a function of λ.
8. Repeat steps 4–7 for a lasso model instead of ridge (i.e., use $\alpha = 1$ in glmnet).
9. Repeat the previous steps with a lasso model, but with smaller values of SNR (e.g. SNR = 0.5, 0.25, 0.1). What do you observe?

Exercise 2: Cross-validation

1. Generate data as in the previous exercise with `SNR = 1.0`.
2. Run `?cv.glmnet` to see what `cv.glmnet` returns.
3. Fit a lasso model using cross-validation:

```
cvlasso <- cv.glmnet(X, y, type.measure="mse",
                      family="gaussian", alpha=1.0)
```

4. Plot the mean cross-validated error as a function of `lambda`.
5. Run `plot(cvlasso)` to plot the cross-validated error and its standard error.
6. Fit a lasso model (no cross-validation) with parameter $\lambda =$ `cvlasso$lambda.min`. Examine the coefficients.

```
best_lasso <- glmnet(X,y, family="gaussian",
                     alpha=1.0, lambda=cvlasso$lambda.min)
```

7. What does the variable `cvlasso$lambda.1se` contain?
8. Get the non-zero coefficients in the previous model: `which(best_lasso$beta != 0)`.
9. Fit a linear model (`lm`) using only the lasso selected variables.

Exercise 3: Breast cancer tumors

The file `Westbc.rda` (available on Sakai) contains gene expression data ($p = 7,129$ genes) for $n = 49$ breast cancer tumor samples (West et al., 2001).

1. Load the data using `load("path-to-file/Westbc.rda")`. (You should have two new variables: `Westbc$assay` and `Westbc$pheno`).
2. Convert the variables `Westbc$pheno` to binary (0/1) values:

```
pheno <- matrix(rep(0,49), nrow=49)
pheno[Westbc$pheno == ’positive’] = 1
```

3. Split the data into a training set $(2/3)$ and a test set $(1/3)$ randomly.
4. Fit a lasso model on the training set using cross-validation.
5. Plot the resulting cross-validation error (`plot(cvlasso)`).
6. Compute the prediction error on the test set using the optimal model.
7. Repeat the previous experiment with 100 random train/test sets and compute the average test error.