Statistics of Big Data, Spring 2018

# Homework 2

Due date: 25 April 2018

1. **Trying out different sparse modeling approaches**

   On the class home page you will find links to two $n >> p$ datasets with $n = 200$ rows and $p = 2000$ columns each, generated from the same sparse linear model, with only four non-zero coefficients (+noise). Our goal is to use the data *train.csv* to build a prediction model with four variables (plus intercept) and then apply this model to *test.csv* to examine its performance.

   (a) Investigate the correlation structure of the columns of the training data. For example, you can sample 1000 pairs of columns $(k_1, k_2)$ and plot their empirical correlation on the y axis and their distance $|k_1 - k_2|$ on the x axis. What do you conclude about the correlation structure? Is it similar to what we discussed in GWAS? Does it comply with compressed sensing assumptions?

   (b) Now we want to apply three sparse modeling scenarios to this data:

      i. *GWAS-like marginal regression:* Apply marginal regression to each column, and choose the best four columns under the constraint that their indexes are at least 70 apart. Then build a linear regression model with the chosen variables only.

      ii. *Relaxed Lasso:* Generate the LARS-Lasso path, stop when it has four variables, then fit a linear regression model with those variables.

      iii. $L_0$ *variable selection:* Find the best "simultaneous" subset of four variables to choose. Since this is practically impossible in 2000 dimensions (at least with current R functions), we will implement it by running a four-variable-selection procedure on the indexes $1-200$, $101-300$, $201-400$ etc. (total of 19 models), then run it once again on the union of variables selected by all models (should be about 50), to obtain a final model. Again, we then build a linear regression model with these 4 variables only.

   Some code examples are given in the file *sparse.r*, pointed from the class home page.

   For each method, present the final model selected, with the variable identities. Also document and compare the total running time the system took to build it (for example using Sys.time() or proc.time() in R).

   (c) Now apply each of these models to the test data and compare their prediction performance.

   (d) How confident are you that you found the correct variables in each approach? Relate this to compressed sensing assumptions and their validity here.

2. **Which properties of Lasso path generalize to other loss functions?**
Recall we showed the optimality conditions for a Lasso solution:

$$\hat{\beta}(\lambda)_k \neq 0 \quad \Rightarrow \quad X_k^T(Y - X\hat{\beta}(\lambda)) = \frac{\lambda}{2}\mathrm{sgn}(\hat{\beta}(\lambda)_k) \tag{1}$$

$$\hat{\beta}(\lambda)_k = 0 \quad \Leftarrow \quad |X_k^T(Y - X\hat{\beta}(\lambda))| < \frac{\lambda}{2} \tag{2}$$

$$\forall k \qquad |X_k^T(Y - X\hat{\beta}(\lambda))| \leq \frac{\lambda}{2}, \tag{3}$$

where as we noted in class,

$$X_k^T(Y - X\hat{\beta}(\lambda)) = -\frac{\partial RSS(\beta)}{\partial \beta_k}|_{\beta=\hat{\beta}(\lambda)}$$

is the derivative of the loss function.

We noted in class the following properties of the set of solutions $\{\hat{\beta}(\lambda) : 0 \leq \lambda \leq \infty\}$:

i All the variables in the solution are "highly correlated" with the current residual from (1) above, and all the variables with zero coefficients are "less correlated" with the current residual from (2,3) above.

ii The solution path $\{\hat{\beta}(\lambda) : 0 \leq \lambda \leq \infty\}$ as a function of $\lambda$ can be described by a collection of "breakpoints" $\infty > \lambda_1 > \lambda_2 > ... > \lambda_K > 0$ such that the set $\mathcal{A}_k$ of active variables with non-zero coefficients is fixed for all solutions $\hat{\beta}(\lambda)$ with $\lambda_k \geq \lambda \geq \lambda_{k+1}$.

iii $\hat{\beta}(\lambda)$ is a piecewise linear function, in other words, for $\lambda$ in this range we have:

$$\hat{\beta}(\lambda) = \hat{\beta}(\lambda_k) + v_k(\lambda_k - \lambda),$$

for a vector $v_k$ we explicitly derived in class.

Assume now that we want to build a different type of model with a different convex and infinitely differentiable loss function, say a logistic regression model for a binary classification task, and add lasso penalty to that:

$$\hat{\beta}(\lambda) = \arg\min_{\beta} \sum_{i=1}^{n} \log\left\{1 + \exp\{-y_i x_i^T \beta\}\right\} + \lambda\|\beta\|_1.$$

We would like to investigate which of the properties above still holds for the solution of this problem.

(a) Using simple arguments about derivatives and sub-derivatives as we used in class for the quadratic loss case, argue that that three conditions like (1)-(3) can be written for this case too, with the appropriate derivative replacing the empirical correlation. Derive these expressions explicitly for the logistic case.

(b) Explain clearly why this implies that properties (i), (ii) still hold (for (ii), you may find the continuity of the derivative useful).

(c) Does the piecewise linearity still hold? A clear intuitive explanation is sufficient here.
**Hint:** Consider how we obtained the linearity for squared loss in $\triangle\lambda$ in class by decomposing the correlation vector $X^T(Y - X\beta) = X^TY - X^TX\beta$.

(d) (* Extra credit) Read the paper "Following Curved Regularized Optimization Solution Paths"[1], also pointed from the class home page, and explain briefly how it proposes to generate the set of solutions to problems like logistic+lasso, and in particular how it takes advantage of the structure (1)-(3).

[1] http://papers.nips.cc/paper/2600-following-curved-regularized-optimization-solution-paths.pdf