MATH 680 Homework 3 Fall 2015

This homework is due on Monday, November 9 at 11:59pm. Submit your solutions (for problems 1--3) in a pdf document on MyCourses. Please also submit text files with your R code (for problem 1--3), which must be commented and properly indented. For problem 4, submit the R package file which should have the suffix .tar.gz.

1. Please see homework 1 for definitions of $X, y, \tilde{X}, \tilde{y}$ and β_*. Suppose we want to compute the bridge penalized least-squares estimate of β_* defined by

$$\hat{\beta}_{-1}^{(\lambda,\alpha)} \in \underset{\tilde{\beta}\in\mathbb{R}^{p-1}}{\arg\min} \left\{ \frac{1}{2}\|\tilde{y} - \tilde{X}\tilde{\beta}\|^2 + \frac{\lambda}{\alpha}\sum_{j=1}^{p-1} |\tilde{\beta}_j|^\alpha \right\} \tag{1}$$
$$\hat{\beta}_1^{(\lambda,\alpha)} = \bar{y} - \bar{x}'\hat{\beta}_{-1}^{(\lambda,\alpha)},$$

where $\lambda \geq 0$ and $\alpha \in [1, 2]$ are tuning parameters. When $\alpha = 2$, (1) becomes ridge-penalized least-squares. When $\alpha = 1$, (1) becomes lasso-penalized least squares.

   (a) Let f be the objective function in (1). Is f convex for $\alpha \in [1, 2]$? Explain.

   (b) Suppose that $\lambda > 0$. Find the set $S_1 \subset [1, 2]$ for which f is differentiable when $\alpha \in S_1$. Find the set $S_2 \subset [1, 2]$ for which f is twice differentiable when $\alpha \in S_2$. Provide justification.

   (c) Create an algorithm to compute $\hat{\beta}^{(\lambda,\alpha)} = (\hat{\beta}_1^{(\lambda,\alpha)}, \hat{\beta}_{-1}^{(\lambda,\alpha)\prime})'$ for $\alpha \in (1, 2)$. Write an R function to test this algorithm. This function should have six arguments:
   - `X`, the design matrix in $\mathbb{R}^{n\times p}$, with ones in its first column.
   - `y`, the response vector in $\mathbb{R}^n$
   - `lam`, this is λ.
   - `alpha`, this is $\alpha \in (1, 2)$.
   - `tol`, this is a value used to determine algorithm convergence.
   - `maxit`, this is the maximum number of iterations allowed.

   and return a list with three elements:
   - `b1`, this is $\hat{\beta}_1^{(\lambda,\alpha)} \in \mathbb{R}$
   - `b`, this is $\hat{\beta}_{-1}^{(\lambda,\alpha)} \in \mathbb{R}^{p-1}$
   - `total.iterations`, this is the number of iterations the algorithm performed.

   (d) Pick reasonable values for n, p, β_*, ω_* and X and randomly generate data from the Normal linear regression cases model to test your function. Use your R function to compute $\hat{\beta}_{\lambda,\alpha}$ for some positive value of λ and some $\alpha \in (1, 2)$. Is the final iterate approximately a stationary point of f?

2. In Binomial logistic regression, we assume that the observed response y_i for the ith case is the observed value of the sample proportion of successes in n_i independent and identical Bernoulli trials where the explanatory variables had values $x_i = (1, x_{i2}, \ldots, x_{ip})'$. Specifically, y_i is a realization of Y_i where

$$n_i Y_i \sim \text{Binom}\left(n_i, \frac{\exp(x_i'\beta_*)}{1 + \exp(x_i'\beta_*)}\right), \quad i = 1, \ldots, n;$$

$Y_1, \ldots, Y_n$ are independent; and $\beta_* \in \mathbb{R}^p$ is the unknown vector of regression coefficients. Then

$$
\begin{aligned}
P(Y_i = y_i) &= P(n_i Y_i = n_i y_i) \\
&= \binom{n_i}{n_i y_i} \left[\frac{\exp(x_i'\beta_*)}{1+\exp(x_i'\beta_*)}\right]^{n_i y_i} \left[1 - \frac{\exp(x_i'\beta_*)}{1+\exp(x_i'\beta_*)}\right]^{n_i - n_i y_i}
\end{aligned}
$$

where $y_i = 0/n_i, 1/n_i, \ldots, n_i/n_i$. The non-random log-likelihood function $\ell : \mathbb{R}^p \to \mathbb{R}$ is defined by

$$
\begin{aligned}
\ell(\beta) = &\sum_{i=1}^{n} n_i y_i \log\left[\frac{\exp(x_i'\beta)}{1+\exp(x_i'\beta)}\right] \\
&+ \sum_{i=1}^{n} (n_i - n_i y_i) \log\left[1 - \frac{\exp(x_i'\beta)}{1+\exp(x_i'\beta)}\right] + \sum_{i=1}^{n} \log\binom{n_i}{n_i y_i}.
\end{aligned}
$$

Define the Bridge-penalized logistic regression estimate of β_* by

$$
\bar\beta^{(\lambda,\alpha)} = \operatorname*{arg\,min}_{\beta \in \mathbb{R}^p} \left\{ -\ell(\beta) + \frac{\lambda}{\alpha} \sum_{j=2}^{p} |\beta_j|^{\alpha} \right\}, \qquad (2)
$$

where $\lambda \geq 0$ and $\alpha \in [1, 2]$ are tuning parameters.

(a) Let g be the objective function in (2). Is g convex for $\alpha \in [1, 2]$? Explain.

(b) Create an algorithm to compute $\bar\beta_\lambda$ for $\alpha \in (1, 2)$. Write an R function to test this algorithm. This function should have six arguments:

- `X`, the model matrix in $\mathbb{R}^{n \times p}$, with ones in its first column.
- `y`, the response vector in $\mathbb{R}^n$, the observed sample proportions.
- `n.list`, a vector of n entries with ith entry n_i.
- `lam`, this is λ.
- `alpha`, this is $\alpha \in (1, 2)$.
- `tol`, this is a value used to determine algorithm convergence.
- `maxit`, this is the maximum number of iterations allowed.

and return a list with two elements:

- `b`, this is the final iterate for $\bar\beta^{(\lambda,\alpha)}$.
- `total.iterations`, this is the number of iterations the algorithm performed.

(c) Pick reasonable values for $n, n_1, \ldots, n_n, p, x_1, \ldots, x_n$, and β_* and generate data from the Binomial logistic regression model. Use your R function to compute $\bar\beta^{(\lambda,\alpha)}$ for some positive value of λ and some value $\alpha \in (1, 2)$. Is this final iterate approximately a stationary point of g?

3. In this problem you will fit a Gaussian mixture model and use it to cluster n observations $x_1, \ldots, x_n$, where $x_i \in \mathbb{R}^p$. The model assumes that $x_1, \ldots, x_n$ are a realization of the sequence of independent random vectors $X_1, \ldots, X_n$ defined in the following way. For each X_i there

is an unobservable random variable Y_i and $(X_1, Y_1), \ldots, (X_n, Y_n)$ are independent copies of (X, Y) where Y is supported on $\{1, \ldots, C\}$ and $(X|Y = y) \sim N_p(\mu_{*y}, \Sigma_{*y})$ for $y \in \{1, \ldots, C\}$. Let $P(Y = y) = \pi_{*y}$ for $y \in \{1, \ldots, C\}$. To fit this model, we estimate

$$\theta_* = (\pi_{*1}, \ldots, \pi_{*C}, \mu_{*1}, \ldots, \mu_{*C}, \Sigma_{*1}, \ldots, \Sigma_{*C}).$$

Let $f_X(;\theta)$ be the density for X with respect to ν_X (lebesgue measure on $\mathbb{R}^p$), let $f_Y(;\theta)$ be the density for Y with respect to ν_Y (counting measure on $\{1, \ldots, C\}$), and let $f_{X|Y}(\cdot; y, \theta)$ be the density of X given $Y = y$ with respect to ν_X, i.e. $f_{X|Y}(\cdot; y, \theta) = \phi(\cdot; \mu_y, \Sigma_y)$ is the density for $N_p(\mu_y, \Sigma_y)$. We have that

$$f_X(x; \theta) = \int f_{X|Y}(x; y, \theta) f_Y(y; \theta) \nu_Y(dy) = \sum_{y=1}^{C} \phi(x, \mu_y, \Sigma_y)\pi_y.$$

So to estimate θ_* with a realization $x_1, \ldots, x_n$ of n independent copies of X, we maximize the loglikelihood function, i.e. we compute a local minimizer for

$$\arg\min_{\theta \in \Theta} - \left[\sum_{i=1}^{n} \log \left\{ \sum_{y=1}^{C} \phi(x_i; \mu_y, \Sigma_y)\pi_y \right\} \right], \tag{3}$$

where $\theta = (\pi_1, \ldots, \pi_C, \mu_1, \ldots, \mu_C, \Sigma_1, \ldots, \Sigma_C)$ and $\Theta = \{r \in (0,1)^C : \sum_{j=1}^{p} r_j = 1\} \times \mathbb{R}^p \times \cdots \times \mathbb{R}^p \times \mathbb{S}_+^p \times \cdots \times \mathbb{S}_+^p$.

(a) Write an R function that uses the EM algorithm to compute a local minimizer for (3). Choose appropriate arguments and return values for the function.

(b) Derive an EM algorithm to compute a local minimizer for

$$\arg\min_{\theta \in \Theta} - \left[\sum_{i=1}^{n} \log \left\{ \sum_{y=1}^{C} \phi(x_i; \mu_y, \Sigma)\pi_y \right\} \right], \tag{4}$$

where $\theta = (\pi_1, \ldots, \pi_C, \mu_1, \ldots, \mu_C, \Sigma)$ and $\Theta = \{r \in (0,1)^C : \sum_{j=1}^{p} r_j = 1\} \times \mathbb{R}^p \times \cdots \times \mathbb{R}^p \times \mathbb{S}_+^p$. The objective function in (4) is the negative loglikelihood function for the model where $(X|Y = y) \sim N_p(\mu_{*y}, \Sigma_*)$, so the covariance matrix is the same for all $y \in \{1, \ldots, C\}$. Write the corresponding R function to compute this local minimizer.

(c) Find a classification dataset from the UCI machine learning data repository for which the Gaussian mixture model assumptions are not terribly unreasonable (pretending that the responses/class labels are unobserved). This repository is at

`https://archive.ics.uci.edu/ml/datasets.html`

Test the R functions created in parts a and b on these data by pretending that the responses/class labels $y_1, \ldots, y_n$ are unobserved. Set C equal to the number possible categories for a response/class label. For a fitted Gaussian mixture model, assign a class label $\hat{y}_i$ to the ith observation x_i with

$$\hat{y}_i = \arg\max_{y \in \{1, \ldots, C\}} \widehat{P}(Y = y | X = x_i),$$

where $\widehat{P}(Y = y|X = x_i)$ is an estimate of $P(Y = y|X = x_i)$ from the Gaussian mixture model fit. Using the fit from the R function created in part a, report $n^{-1} \sum_{i=1}^{n} 1(\hat{y}_i = y_i)$. Also do this for the fit from the R function created in part b.

4. Create an R package called `mylasso` that allows the user to call two functions: `minlassoLSwithC` and `minlassoLS`, where these functions are defined in "rlasso.r" posted on moodle.

    - The definition of `minlassoLSwithC` requires the C function defined in "rlasso.c" that is also posted on moodle.
    - This package should have three help files: one for the function `minlassoLSwithC`, one for the function `minlassoLS`, and one for the R package `mylasso`.
    - Build the package on a linux machine using `R CMD build`. Linux machines are available in 352 Ford Hall and could be accessed on your personal computer using an ssh client.
    - Check the package with `R CMD check` and make sure that it passes all of the tests (copy and paste the output).
    - Install your built package using `R CMD INSTALL` and make sure that the functions work properly and that the help files are displaying properly.
    - Upload the built package file, which should have a suffix of .tar.gz