**550.740 Project 3; Optimal Scoring and Lasso.**
**Due on April 15 2019**

- **Please type your answers (using, e.g., LateX or MS Word.)**
- *The solution must be written with sufficient explanations and your results must be commented. Returning a series of numerical results and figures is not enough. A solution in the form of a program output is not acceptable either.*
- *Please return your program sources. They will not graded (so no direct credit for them), but they will be useful in order to understand why results are not correct (and decide whether partial credit can be given) and to ensure that your work is original. You may use any programming language, although Python, Matlab or R are recommended.*
- *The ".csv" files associated with this project are available on Blackboard.*

**Question 1.**

Given a training set $(x_1, y_1), \ldots, (x_N, y_N)$ with $x_k \in \mathbb{R}^d$ and $y_k \in \mathcal{G} = \{g_1, \ldots, g_q\}$, optimal scoring is formulated in terms of the minimization of

$$F_0(\theta, b) = -2\text{trace}(b^T M^T C\theta) + \text{trace}(b^T \Sigma_{XX} b)$$

in $\theta$ and $b$ with the constraints $\theta^T C\theta = \text{Id}_{\mathbb{R}^r}$ and $\theta^T C\mathbf{1}_d = 0$. Here, $b$ is a $d \times r$ matrix;

$$\theta = \begin{pmatrix} \theta_{g_1}^T \\ \vdots \\ \theta_{g_q}^T \end{pmatrix}$$

is a $q \times r$ matrix (with $r$ chosen between 1 and $q-1$);

$$M = \begin{pmatrix} (\mu_{g_1} - \mu)^T \\ \vdots \\ (\mu_{g_q} - \mu)^T \end{pmatrix}$$

is a $q \times d$ matrix, where $\mu_g$ is the average over $x_k$ such that $y_k = g$ and $\mu$ is the global average over all $x_k$'s; $C$ is the diagonal matrix with diagonal coefficients $N_{g_1}/N, \ldots, N_{g_q}/N$, $N_g$ being the size of class $g$ and $\mathbf{1}_d$ is the $d$-dimensional vector with all coefficients equal to one; finally, $\Sigma_{XX} = \mathcal{X}_c^T \mathcal{X}_c / N$, where $\mathcal{X}_c$ is as usual the $N$ by $d$ matrix with $k$th row given by $(x_k - \mu)^T$.

In this question we consider the problem of minimizing $F_0$ with respect to $\theta$ for fixed $b$. Prove that, if $\text{rank}(Mb) \geq r$, the optimal $\theta$ is given by

$$\theta = C^{-1/2} U \begin{pmatrix} V^T \\ 0 \end{pmatrix}.$$

where $UDV^T$ is the SVD decomposition of $C^{1/2}Mb$. (Hint: reformulate the problem in terms of $\tilde{\theta} = U^T C^{1/2}\theta V$.)

**Question 2.**

We now consider a penalized version of the previous problem minimizing

$$F_\lambda(\theta, b) = -2\text{trace}(b^T M^T C\theta) + \text{trace}(b^T \Sigma_{XX} b) + \lambda \sum_{i=1}^{d}\sum_{j=1}^{r} |b(i,j)|$$

in $\theta$ and $b$ with the constraints $\theta^T C\theta = \text{Id}_{\mathbb{R}^r}$ and $\theta^T C\mathbf{1} = 0$. We now focus on the solution of the problem with fixed $\theta$, i.e., the minimization in $b$.

(1) Prove (using results stated in class) that the optimal $b$ can be obtained using the following version of the ADMM algorithm, iterating

$$\begin{cases} b \leftarrow \left(\Sigma_{XX} + \dfrac{\text{Id}}{2\rho}\right)^{-1} \left(M^T C\theta + (\gamma - \tau)/(2\rho)\right) \\ \gamma \leftarrow S_{\lambda\rho}\left(b + \tau\right) \\ \tau \leftarrow \tau + b - \gamma \end{cases}$$

Here $\tau$ and $\gamma$ are $d \times r$ matrices, $\rho$ is a small enough positive number and $S_{\lambda\rho}$ is the shrinking operator, applying the function $t \mapsto (|t| - \lambda\rho)^+$ to every element of a matrix.

Hint: You can rewrite the minimization in $b$ with a fixed $\theta$ as $r$ independent problems (one for each column of $b$) that are similar to lasso and apply the ADMM algorithm described in class to each of them.

(2) Completely describe a minimization algorithm for $F_\lambda$ that is initialized with $\theta = C^{-1/2}\begin{pmatrix}\text{Id}_{\mathbb{R}^r} \\ 0\end{pmatrix}$ and alternates a minimization step with fixed $\theta$ and a minimization step with fixed $b$ until the numerical convergence.

**Question 3.**

(1) Program a training function taking as input:

- A training set, in the form of an $N \times d$ matrix $X$ and an $N \times 1$ class vector $Y$.
- The number of scores, $r$.
- The penalty parameter $\lambda$.
- A maximal number of iterations

The function should minimize $F_\lambda$ using the algorithm described in question 2.2 applied to the training set $X$ in which each coordinate (column) has been divided by its standard deviation.

The program should return the optimal $b$ and $\theta$, the global average, $\mu$. You need to "unnormalize" your result so that $b$ can be applied to the original data (i.e., appropriately divide the rows of $b$ by the standard deviations of the columns of $X$).

When performing the minimization in $\theta$ your program should test that $\text{rank}(Mb) \geq r$ and stop with an error if this happens during the procedure.

Test your program with training and test data in "Train_project3_Q3.csv" and "Test_project3_Q3.csv" for which $d = 5$ and $q = 2$ (the first five columns of this file contain the $x$ variables and the last one is the class. Using $\lambda = 5/\sqrt{N}$, provide the training and test errors and the number of non-zero coefficients obtained for $b$.

(2) Run your algorithm on the training and test datasets "Train_project3_Q3_2.csv" and "Test_project3_Q3_2.csv" (for which $d = 30$ and $q = 3$) with $r = 2$ and for $\sqrt{N}\lambda = 0.1$, 0.5, 1, 2.5, 5, 7.5, 10, 20 and 30. For each $\lambda$, return:

- The total number of iterations of the alternating procedure above that were required.
- A plot of the cost function after each minimization step, making sure it non-increasing. (Provide this only for $\sqrt{N}\lambda = 0.1$, 5 and 10.)
- The indices of non-zero coefficients of each column of $b$.
- The rates of correct classification on the training set and on the test set.