Final Exam, CPSC 8420, Fall 2020

Last Name, First Name

Due 12/11/2020, Friday, 11:59PM EST

Problem 1

[15 pts] Consider the elastic-net optimization problem:

$$\min_\beta \|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda[\alpha\|\beta\|_2^2 + (1-\alpha)\|\beta\|_1]. \tag{1}$$

1. Show the objective can be reformulated into a lasso problem, with a slightly different $\hat{\mathbf{X}}, \hat{\mathbf{y}}$.

2. If we fix $\alpha = .5, \lambda = 1$, please derive the closed solution by making use of alternating minimization that each time we fix the rest by optimizing one single element in β. You need write a program, randomly generate $\mathbf{X}, \mathbf{y}$ and initialize β_0, then show that the objective decreses monotonically with updates.

1. $\hat{\mathbf{X}} = \begin{bmatrix} \mathbf{X} \\ \sqrt{\lambda\alpha}\mathbf{I} \end{bmatrix}, \hat{\mathbf{y}} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix}$, *then we have:*

$$\|\hat{\mathbf{y}} - \hat{\mathbf{X}}\beta\|^2 = \left\|\begin{bmatrix} \mathbf{y} - \mathbf{X}\beta \\ -\sqrt{\lambda\alpha}\beta \end{bmatrix}\right\|^2 = \|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda\alpha\|\beta\|^2 \tag{2}$$

thus it is equivalent to solve:

$$\min_\beta \|\hat{\mathbf{y}} - \hat{\mathbf{X}}\beta\|^2 + \lambda(1-\alpha)\|\beta\|_1,$$

which is a Lasso problem.

2. Let's consider vanilla Lasso as:

$$\min_\beta \frac{1}{2}\|\mathbf{y} - \mathbf{X}\beta\|^2 + \theta\|\beta\|_1$$

it is equivalent to

$$\min_{\beta_j} \frac{1}{2}\|\underbrace{\mathbf{y} - \sum_{i\neq j} \mathbf{x}_i\beta_i}_{\mathbf{y}_i} - \mathbf{x}_j\beta_j\|^2 + \theta|\beta_j| = \frac{1}{2}\|\mathbf{x}_j\beta_j - \mathbf{y}_i\|^2 + \theta|\beta_j|, \forall j$$

$$\beta_j = \begin{cases} \frac{\langle \mathbf{x}_j, \mathbf{y}_i\rangle - \theta}{\|\mathbf{x}_j\|^2} & \textit{if} \quad \langle \mathbf{x}_j, \mathbf{y}_i\rangle > \theta, \\ \frac{\langle \mathbf{x}_j, \mathbf{y}_i\rangle + \theta}{\|\mathbf{x}_j\|^2} & \textit{if} \quad \langle \mathbf{x}_j, \mathbf{y}_i\rangle < -\theta, \\ 0 & \textit{else.} \end{cases} \tag{3}$$

Problem 2

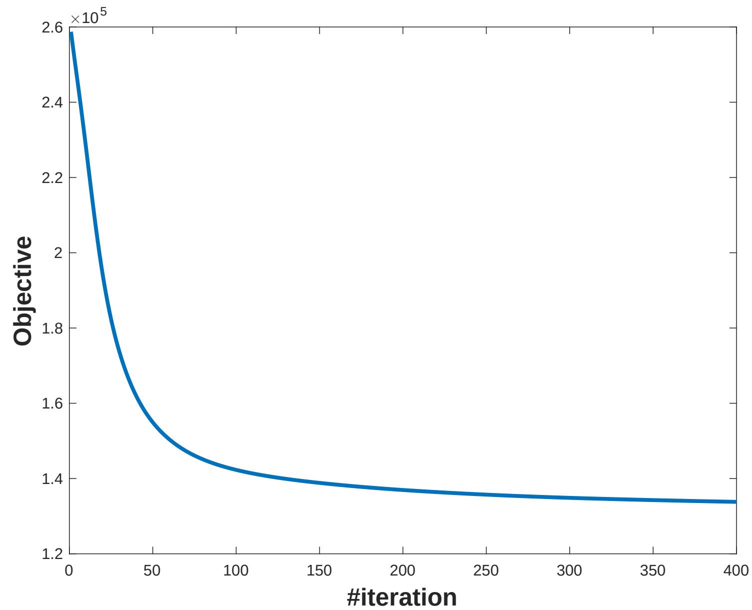
[15 pts] Following the idea in Non-negative Matrix Factorization (NMF), please propose an updating algorithm to optimize:

$$\min_{\mathbf{F},\mathbf{G}\geq 0} \|\mathbf{X}-\mathbf{F}\mathbf{G}\|^2 + \lambda\, \mathbf{tr}\left(\mathbf{G}\mathbf{L}\mathbf{G}^T\right) \tag{4}$$

where $\mathbf{X} \in \mathbb{R}^{m\times n}$, with each column denotes a data, $\mathbf{F} \in \mathbb{R}^{m\times r}, \mathbf{G} \in \mathbb{R}^{r\times n}$; $\mathbf{L} := \mathbf{D} - \mathbf{W}$ is the Laplacian matrix where $\mathbf{W}(i,j) := exp(-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{2})$ and $\mathbf{D}$ is degree matrix which is diagonal $\mathbf{D}(i,i) = \sum_{j=1}^{n} \mathbf{W}(i,j)$. Then please write a program to illustrate your algorithm decreases the objective monotonically (where you may set $\lambda = 1$) within each iteration. (you need randomly generate and initialize $\mathbf{X}, \mathbf{F}, \mathbf{G}$.)

$$\begin{aligned}
\mathbf{F}_{k+1} &= \mathbf{F}_k \odot \frac{\mathbf{X}\mathbf{G}_k^T}{\mathbf{F}_k\mathbf{G}_k\mathbf{G}_k^T} \\
\mathbf{G}_{k+1} &= \mathbf{G}_k \odot \frac{\mathbf{F}_{k+1}^T\mathbf{X} + \lambda(\mathbf{G}_k\mathbf{L})^-}{\mathbf{F}_{k+1}^T\mathbf{F}_{k+1}\mathbf{G}_k + \lambda(\mathbf{G}_k\mathbf{L})^+}
\end{aligned} \tag{5}$$

```
set(gca, 'LineWidth' , 1.5,'FontSize',6);
X=20*rand(100,80);
F=10*rand(100,20);
G=2*rand(20,80);
W = zeros(80,80);
for i=1:80
    for j=1:80
        dist=norm(X(:,i)-X(:,j));
        W(i,j)=exp(-dist*dist/2);
    end
end
D = diag(sum(W));
L=D-W;
itr=400;
obj=zeros(itr,1);
for k=1:itr
    F=F.*((X*G')./(F*G*G'));
    POS=(abs(G*L)+G*L)/2;
    NEG=(abs(G*L)-G*L)/2;
    G=G.*((F'*X+NEG)./(F'*F*G+POS));
    obj(k)=power(norm(X-F*G,'fro'),2)+trace(G*L*G');
end
figure
plot(obj,'LineWidth',2.5,'MarkerSize',20)
xlabel('#iteration','FontSize',16,'FontWeight','bold')
ylabel('Objective','FontSize',16,'FontWeight','bold')
```



Problem 3

[10 pts] Recall the Least Squares problem, now assume we have multiple outputs $\mathbf{Y} = [\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_K] \in \mathbb{R}^{N\times K}$ that we wish to predict from inputs $\mathbf{X} \in \mathbb{R}^{N\times(p+1)}$, with $\mathbf{Y} = \mathbf{XB} + \mathbf{E}$, where $\mathbf{B}$ is the parameter matrix to optimize and $\mathbf{E}$ is martix of errors with each element has an expectation value of 0. Now please determint the optimal $\mathbf{B}$ following Least Squares and ridge regression version solution $\mathbf{B}_\lambda^{ridge}$.

$$\min_{\mathbf{B}} \|\mathbf{Y} - \mathbf{XB}\|_F^2 \tag{6}$$

where the solution $\mathbf{B}^{LS} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$

$$\min_{\mathbf{B}} \|\mathbf{Y} - \mathbf{XB}\|_F^2 + \lambda\|\mathbf{B}\|_F^2 \tag{7}$$

where the solution $\mathbf{B}_\lambda^{ridge} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y}$

Problem 4

[10 pts] Let's compare Ridge Regression with vanilla Least Squares:

1. What's the cons of Ridege Regression?

   parameter tuning

2. What's its pros in terms of:

   (a) Bias-Variance tradeoff?
   more accurate w.r.t. test error

   (b) When there are more feature dimensions than data points?
   to guarantee unique solution

   (c) Stability when optimize with closed-form solution?
   to make the condition number smaller

   (d) Convergence rate if leveraging gradient descent to obtain the solution?
   faster as it is proportional to conditional number

Problem 5

[10 pts] Please deterine the optimal solutions with proof:

1. If $\mathbf{x}$ is a vector and $\mathbf{v}$ is known, then please optimize $\min_{\mathbf{x}} \|\mathbf{x} - \mathbf{v}\|^2, \quad s.t. \quad \|\mathbf{x}\|_0 \leq k$.

   to remain the k largest magnitude (positive/negative) number in $\mathbf{x}$, and set the rest to be 0

2. If $\mathbf{X}$ is a matrix and $\mathbf{V}$ is known, then please optimize $\min_{\mathbf{X}} \|\mathbf{X} - \mathbf{V}\|_F^2, \quad s.t. \quad rank(\mathbf{X}) \leq k$. (*hint*: you may refer to PCA/SVD part.)

   assume $[\mathbf{U}, \mathbf{S}, \mathbf{L}] = svd(\mathbf{V})$, *then* $\mathbf{X} = \mathbf{U}(:, 1:k) * \mathbf{S}(1:k) * \mathbf{L}(:, 1:k)^T$

Problem 6

[10 pts] Assume that in a community, there are 10% people suffer from COVID. Tests identifies that 80% of the patients come to breathing difficulty while 25% of those free from COVID also have symptons of shortness of breath. Now please determine that if one has breathing difficulty, what's his/her chance to have COVID? (*hint*: you may consider Naive Bayes)

Denote b as having breathing difficulty, c as suffering from covid, then:

$$
\begin{aligned}
p(c/b) &= \frac{p(c)p(b/c)}{p(b)} \\
p(\bar{c}/b) &= \frac{p(\bar{c})p(b/\bar{c})}{p(b)}
\end{aligned} \tag{8}
$$

thus we have:

$$
\begin{aligned}
&\frac{p(c/b)}{p(\bar{c}/b)} = \frac{p(c)p(b/c)}{p(\bar{c})p(b/\bar{c})} = \frac{.1 * .8}{.9 * .25} \\
&p(c/b) + p(\bar{c}/b) = 1
\end{aligned} \tag{9}
$$

where we can determine $p(c/b) = \frac{16}{61} = 26.23\%$

Problem 7

[15 pts] Given m training examples $(\mathbf{x}_i, \mathbf{x}_j)(i, j = 1, \ldots, m)$, the kernel matrix $\mathbf{A}(i, j) = K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \langle \mathbf{x}_i, \mathbf{x}_j \rangle)^2$. Prove $\mathbf{A}$ is semi-positive definite matrix. (*hint*: you may refer to Kernel SVM slides)

Let $\Phi(\mathbf{x}_i)$ be the feature map for the i-th example and define the matrix $\mathbf{B} = [\Phi(\mathbf{x}_1), \ldots, \Phi(\mathbf{x}_m)]$, then we have $\mathbf{A} = \mathbf{B}^T\mathbf{B}$, thus $\mathbf{g}'\mathbf{A}\mathbf{g} = (\mathbf{B}\mathbf{g})^T(\mathbf{B}\mathbf{g}) = \|\mathbf{B}\mathbf{g}\|^2 \geq 0$

Problem 8

[15 pts] [**Open Problem**] How to determine the super-spreaders of COVID and their transmission factor/feature? (you are provided with whatever data you want to use/collect and any machine learning method is acceptable as long as reasonable)