CS/ECE/ME532 Activity 17

Estimated Time: 25 min for P1, 15 min for P2, 25 min for P3

1. **Alternative regularization formulas.** This problem is about two alternative ways of solving the L_2-regularized least squares problem.

   a) Prove that for any $\lambda > 0$, the following matrix identity holds:

   $$(\boldsymbol{A}^T\boldsymbol{A} + \lambda\boldsymbol{I})^{-1}\boldsymbol{A}^T = \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{A}^T + \lambda\boldsymbol{I})^{-1}$$

   Hint: Start by considering the expression $\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{A}^T + \lambda\boldsymbol{A}^T$ and factor it in two different ways (from the right or from the left).

   b) The identity proved in part **a)** shows that there are actually two equivalent formulas for the solution to the L_2-regularized least squares problem. Suppose $\boldsymbol{A} \in \mathbb{R}^{8000\times 100}$ and $\boldsymbol{y} \in \mathbb{R}^{8000}$, and use this identity to find $\boldsymbol{w}$ that minimizes $||\boldsymbol{A}\boldsymbol{w} - \boldsymbol{y}||_2^2 + \lambda||\boldsymbol{w}||_2^2$ in two different ways. Which formula will compute more rapidly? Why? *Note:* The number of operations required for matrix inversion is proportional to the cube of the matrix dimension.

   c) A breast cancer gene database has approximately 8000 genes from 100 subjects. The label y_i is the disease state of the ith subject (+1 if no cancer, -1 if breast cancer). Suppose we build a linear classifier that combines the 8000 genes, say $\boldsymbol{g}_i, i = 1, 2, \ldots, 100$ to predict whether a subject has cancer $\hat{y}_i = \text{sign}\{\boldsymbol{g}_i^T\boldsymbol{w}\}$. Note that here $\boldsymbol{g}_i$ and $\boldsymbol{w}$ are 8000-by-1 vectors.

      i. Write down the least squares problem for finding classifier weights $\boldsymbol{w}$ given 100 labels. Does this problem have a unique solution?

      ii. Write down a Tikhonov(ridge)-regression problem for finding the classifier weights given 100 labels. Does this problem have a unique solution? Which form of the identity in part **a)** leads to the most computationally efficient solution for the classifier weights?

2. The key idea behind proximal gradient descent is to reformulate the general regularized least-squares problem into a set of simpler scalar optimization problems. Consider the regularized least-squares problem

   $$\min_{\boldsymbol{w}} ||\boldsymbol{z} - \boldsymbol{w}||_2^2 + \lambda r(\boldsymbol{w})$$

   An upper bound and completing the square was used to simplify the generalized least-squares problem into this form. Let the i^{th} elements of $\boldsymbol{z}$ and $\boldsymbol{w}$ be z_i and w_i, respectively.

a) Assume $r(\boldsymbol{w}) = ||\boldsymbol{w}||_2^2$. Write the regularized least-squares problem as a series of separable problems involving only w_i and z_i.

b) Assume $r(\boldsymbol{w}) = ||\boldsymbol{w}||_1$. Write the regularized least-squares problem as a series of separable problems involving only w_i and z_i.

3. A script is available to compute a specified number of iterations of the proximal gradient descent algorithm for solving a Tikhonov-regularized least squares problem

$$\min_{\boldsymbol{w}} ||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}||_2^2 + \lambda||\boldsymbol{w}||_2^2$$

The provided script will get you started displaying the path taken by the weights in the proximal gradient descent iteration superimposed on a contour plot of the squared error surface. Assume $\boldsymbol{y} = \begin{bmatrix} \sqrt{2} \\ 0 \\ 1 \\ 0 \end{bmatrix}$, the 4-by-2 $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ has singular value decomposition $\boldsymbol{U} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$, $\boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0 \\ 0 & 1/2 \end{bmatrix}$, and $\boldsymbol{V} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$. Complete 20 iterations of gradient descent in each case specified below.

Include the plots you generate below with your submission.

a) What is the maximum value for the step size τ that will guarantee convergence?

b) Start proximal gradient descent from the point $\boldsymbol{w} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$ using a step size of $\tau = 0.5$ and tuning parameter $\lambda = 0.5$. How do you explain the trajectory the weights take toward the optimum, e.g., why is it shaped this way? What direction does each iteration move in the regularization step?

c) Repeat the previous case with $\lambda = 0.1$ What happens? How does λ affect each iteration and why?

1a.) $(\underline{A}^T\underline{A} + \lambda \underline{I})^{-1}\underline{A}^T = \underline{A}^T(\underline{A}\,\underline{A}^T + \lambda \underline{I})^{-1}$ if $\lambda > 0$.

$\underline{A}^T\underline{A}\,\underline{A}^T + \lambda \underline{A}^T = \underline{A}^T(\underline{A}\,\underline{A}^T + \lambda \underline{I}) = (\underline{A}^T\underline{A} + \lambda \underline{I})\underline{A}^T$

$\rightarrow (\underline{A}^T\underline{A} + \lambda \underline{I})^{-1}\underline{A}^T(\underline{A}\,\underline{A}^T + \lambda \underline{I}) = \underline{A}^T$ LHS

$\rightarrow (\underline{A}^T\underline{A} + \lambda \underline{I})^{-1}(\underline{A}^T\underline{A}\,\underline{A}^T + \lambda \underline{I}) = \underline{A}^T$

$\rightarrow (\underline{A}^T\underline{A} + \lambda \underline{I})^{-1}(\underline{A}^T\underline{A}\,\underline{A}^T + \lambda \underline{A}^T) = \underline{A}^T$

$\rightarrow \cancel{(\underline{A}^T\underline{A} + \lambda \underline{I})^{-1}(\underline{A}^T\underline{A} + \lambda \underline{I})}\underline{A}^T = \underline{A}^T$

$\rightarrow \underline{I}\,\underline{A}^T = \underline{A}^T$ ✓

$(\underline{A}^T\underline{A} + \lambda \underline{I})A^T(\underline{A}\,\underline{A}^T + \lambda \underline{I})^{-1} = \underline{A}^T$

$\rightarrow (\underline{A}^T\underline{A} + \lambda \underline{I})A^T(\underline{A}\,\underline{A}^T + \lambda \underline{I})^{-1} = \underline{A}^T$

$\rightarrow (\underline{A}^T\underline{A}\,\underline{A}^T + \underline{A}^T\lambda)(\underline{A}\,\underline{A}^T + \lambda \underline{I})^{-1}$

$\rightarrow \underline{A}^T\cancel{(\underline{A}\,\underline{A}^T + \lambda \underline{I})(\underline{A}\,\underline{A}^T + \lambda \underline{I})^{-1}} = A^T$

$\rightarrow \underline{A}^T\underline{I} = \underline{A}^T$ ✓

1b.) $\underline{A}$, 8000×100, y: 8000×1

$$(\underline{A}^T\underline{A} + \lambda \underline{I})^{-1}\underline{A}^T = \underline{A}^T(\underline{A}\,\underline{A}^T + \lambda \underline{I})^{-1}$$

$$\underbrace{\underline{A}^T\underline{A} \text{ is } 100 \times 100}_{\text{This expression will invert much faster.}} \qquad \underline{A}\,\underline{A}^T \text{ is } 8000 \times 8000$$

1c.) i.) $\underline{w}_{min} = (\underline{G}^T\underline{G})^{-1}\underline{G}^T y$

$$\underset{8000 \times 100}{\begin{array}{c} \\ 8000 \end{array}\begin{array}{c} 100 \\ \begin{bmatrix} \underline{G} \end{bmatrix} \end{array}} \underset{100 \times 1}{\begin{bmatrix} \underline{w} \end{bmatrix}} = \underset{8000 \times 1}{\begin{bmatrix} y \end{bmatrix}}$$

$$\min_{\underline{w}} \|\underline{G}\underline{w} - y\|_2^2$$

If rank $\underline{G}$ = rank $(\underline{G} \vdots y)$

& rank G = dim$(\underline{w})$ = 100, then there is a unique solution.

ii) $\min_{\underline{w}} \|\underline{G}\underline{w} - \underline{y}\|_2^2 + \lambda\|\underline{w}\|_2^2$.

$$\rightarrow \underline{w}_{min} = \underline{G}^T(\underline{G}\,\underline{G}^T + \lambda \underline{I})^{-1} y = \underbrace{(\underline{G}^T\underline{G} + \lambda \underline{I})^{-1}\underline{G}^T y}$$

$\underline{G}^T\underline{G}$ is 100×100

will invert much faster.

2.) $\min\limits_{\underline{w}} \| \underline{z} - \underline{w} \|_2^2 + \lambda\, r(\underline{w})$

$f(\underline{w}) = \| \underline{z} - \underline{w} \|_2^2 + \lambda\, r(\underline{w})$

a.) If $r(\underline{w}) = \| \underline{w} \|_2^2$

$\rightarrow \quad r(\underline{w}) = \sum_{i=1}^{m} w_i^{\,2}$.

$\rightarrow \quad \min\limits_{\underline{w}} \| \underline{z} - \underline{w} \|_2^2 + \lambda \sum_{i=1}^{m} w_i^{\,2} \rightarrow \min\limits_{w_i} \sum_i \sqrt{(z_i^2 + \underline{w}_i^2)} + \lambda w_i^{\,2}$

b.) $\min\limits_{w_i} \sum_i \sqrt{(z_i^2 + \underline{w}_i^2)} + \lambda |\underline{w}_i|$

3a.) $\tau \leq 1 / \| \underline{A} \|_{op} = 1/\delta_1 = 1.$

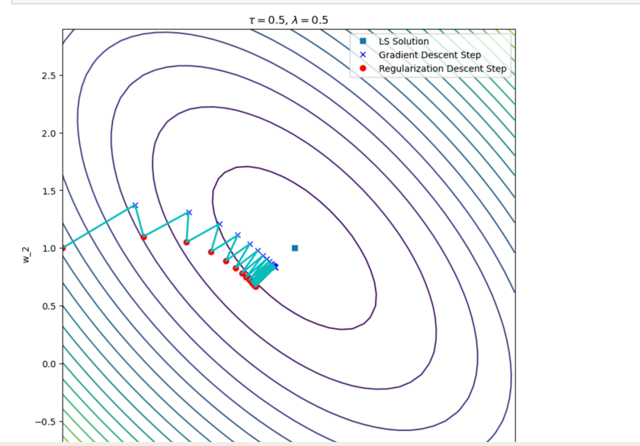
3b.)

Each regularization step pulls the trajectory at the direction perpendicular to the surface.

```python
w_init = np.array([[-1],[1]])
lam = 0.5;
it = 20
tau = 0.5
W,Z = prxgraddescent_l2(X,y,tau,lam,w_init,it)

# Concatenate gradient and regularization steps to display trajectory
G = np.zeros((2,0))
for i in range(it):
    G = np.hstack((G,np.hstack((W[:,[i]],Z[:,[i+1]]))))

plt.figure(figsize=(9,9))
plt.contour(w1,w2,fw,20)
plt.plot(w_ls[0],w_ls[1],"s", label="LS Solution")
plt.plot(Z[0,1::],Z[1,1:],'bx',linewidth=2, label="Gradient Descent Step")
plt.plot(W[0,:],W[1,:],'ro',linewidth=2, label="Regularization Descent Step")
plt.plot(G[0,:],G[1,:],'-c',linewidth=2)
plt.legend()
plt.xlabel('w_1')
plt.ylabel('w_2')
plt.title('$\\tau = $'+str(.5)+', $\lambda = $'+str(lam));
```



3c)

As $\lambda \downarrow$,

the trajectory is pulled "less hard" in the $\perp$ direction by the regularizing term.

This allows us to get closer to LS solution ultimately.

```
for i in range(it):
    G = np.hstack((G,np.hstack((W[:,[i]],Z[:,[i+1]]))))

plt.figure(figsize=(9,9))
plt.contour(w1,w2,fw,20)
plt.plot(w_ls[0],w_ls[1],"s", label="LS Solution")
plt.plot(Z[0,1::],Z[1,1:],'bx',linewidth=2, label="Gradient Descent Step")
plt.plot(W[0,:],W[1,:],'ro',linewidth=2, label="Regularization Descent Step")
plt.plot(G[0,:],G[1,:],'-c',linewidth=2)
plt.legend()
plt.xlabel('w_1')
plt.ylabel('w_2')
plt.title('$\\tau = $'+str(.5)+', $\lambda = $'+str(lam));
```