CS/ECE/ME532 Period 17 Activity

Estimated Time: 25 min for P1, 15 min for P2, 25 min for P4

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?

SOLUTION:

a) Factoring the expression from the hint in two different ways, we have:

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

The terms in parentheses are always invertible, so we can multiply on the right by $(\boldsymbol{A}\boldsymbol{A}^T + \lambda\boldsymbol{I})^{-1}$ and on the left by $(\boldsymbol{A}^T\boldsymbol{A} + \lambda\boldsymbol{I})^{-1}$ to we obtain the desired identity.

b) The two equivalent formulas for the solution to the L_2-regularized least squares problem are:

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

The bottleneck in computation is the matrix inversion. Since $\boldsymbol{A} \in \mathbb{R}^{8000\times 100}$, this means $(\boldsymbol{A}^T\boldsymbol{A} + \lambda\boldsymbol{I}) \in \mathbb{R}^{100\times 100}$, whereas $(\boldsymbol{A}\boldsymbol{A}^T + \lambda\boldsymbol{I}) \in \mathbb{R}^{8000\times 8000}$. So the second formula is on the order of 80^3 slower to compute.

c) Let $\boldsymbol{A} = \begin{bmatrix} \boldsymbol{g}_1^T \\ \boldsymbol{g}_2^T \\ \dots \\ \boldsymbol{g}_{100}^T \end{bmatrix}$ and $\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_{100} \end{bmatrix}$.

i. $\min_{\boldsymbol{w}} ||\boldsymbol{A}\boldsymbol{w} - y||_2^2$. Since $\boldsymbol{A} \in \mathbb{R}^{100\times 8000}$, $\boldsymbol{A}$ has rank at most 100 which is much less than the number of weights $\boldsymbol{w}$. No unique solution.

ii. $\min_{\boldsymbol{w}} ||\boldsymbol{A}\boldsymbol{w} - y||_2^2 + \lambda||\boldsymbol{w}||_2^2$ always has a unique solution given by $\boldsymbol{w} = (\boldsymbol{A}^T\boldsymbol{A} + \lambda\boldsymbol{I})^{-1}\boldsymbol{A}^T\boldsymbol{y}$. However, this requires inverting an 8000 by 8000 matrix, so it is much more efficient to use the above identity and obtain the solution as $\boldsymbol{w} = \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{A}^T + \lambda\boldsymbol{I})^{-1}\boldsymbol{y}$ as this expression only requires inverting a 100 by 100 matrix.

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.

SOLUTION:

a) $||\boldsymbol{w}||_2^2 = \sum_{i=1}^n w_i^2$, so the regularized least-squares problem is

$$\min_{\boldsymbol{w}} \sum_{i=1}^{n} (z_i - w_i)^2 + \lambda w_i^2$$

which is equivalent to

$$\min_{w_i}(z_i - w_i)^2 + \lambda w_i^2, i = 1, 2, \ldots, n$$

b) In this case $||\boldsymbol{w}||_1 = \sum_{i=1}^{n} |w_i|$ so the regularized least-squares problem is

$$\min_{\boldsymbol{w}} \sum_{i=1}^{n}(z_i - w_i)^2 + \lambda|w_i|$$

which is equivalent to

$$\min_{w_i}(z_i - w_i)^2 + \lambda|w_i|, i = 1, 2, \ldots, n$$

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.

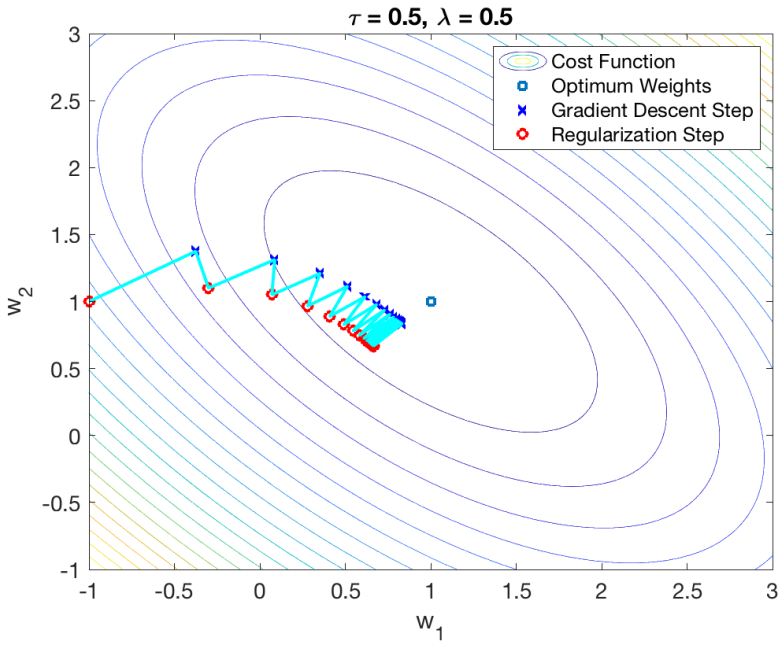
a) What is the maximum value for the step size τ that will guarantee convergence?

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?

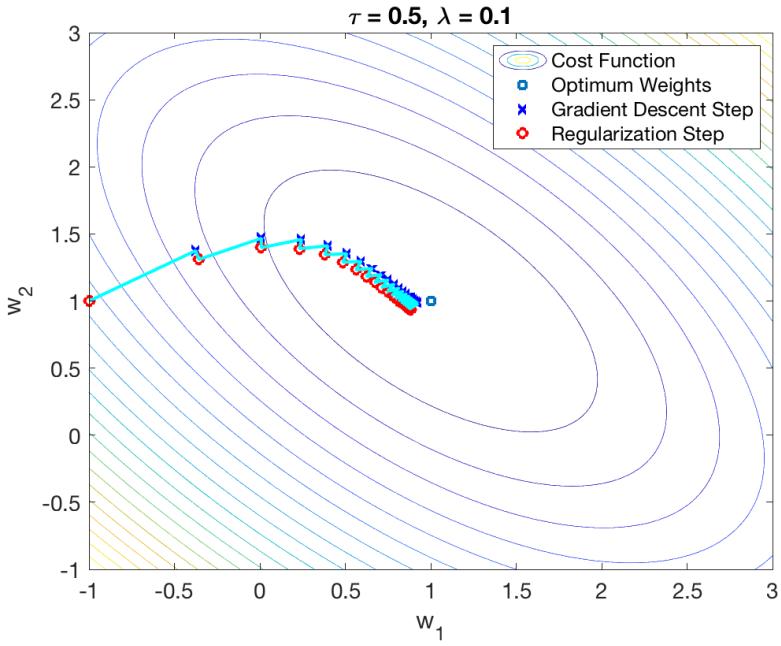
SOLUTION:

a) The proximal gradient descent algorithm is derived assuming $\tau < 1/||\boldsymbol{X}||_{op}^2$ which implies $\tau < 1$ in this problem.



b)

Tikhonov regularization penalizes the length of the solution, so application of the regularizer forces the most recent iteration, marked by the blue x, toward the origin. Then the gradient at that point is used to determine the direction of the next update, that update is shrunk towards the origin, etc. Note that the solution does not converge to the bottom of the least-squares cost function because the regularizer pulls it towards the origin (smaller norm).



c)

Decreasing λ places less emphasis on the regularizer and results in less shrinkage toward the origin at each step. Conversely, if we increased λ then we'd be weighting the norm of the solution more heavily and there would be greater shrinkage of each iteration towards the origin.