Machine Learning, 2024 Spring
Assignment 6

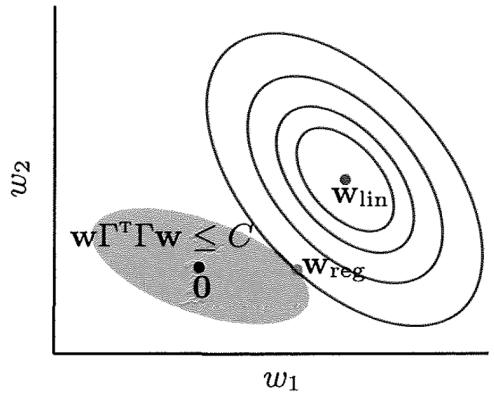
Notice

Plagiarizer will get 0 points.
LaTeXis highly recommended. Otherwise you should write as legibly as possible.

Problem 1 In this problem, you will investigate the relationship between the soft order constraint and the augmented error. The regularized weight $\mathbf{w}_{\text{reg}}$ is a solution to

$$\min E_{\text{in}}(\mathbf{w})$$
$$\text{subject to } \mathbf{w}^{\mathrm{T}}\Gamma^{\mathrm{T}}\Gamma\mathbf{w} \leq C$$

(a) If $\mathbf{w}_{\text{lin}}^{\mathrm{T}}\,\Gamma^{\mathrm{T}}\Gamma\mathrm{w}_{\text{lin}} \leq C$, then what is $\mathbf{w}_{\text{reg}}$? (5pt)
(b) If $\mathbf{w}_{\text{lin}}^{\tau}\,\Gamma^{\top}\Gamma\mathbf{w}_{\text{lin}} > C$, the situation is illustrated below,



The constraint is satisfied in the shaded region and the contours of constant E_{in} are the ellipsoids (why ellipsoids?). What is $\mathbf{w}_{\text{reg}}^{\mathrm{T}}\,\Gamma^{\mathrm{T}}\Gamma\mathbf{w}_{\text{reg}}$? (5pt)
(c) Show that with

$$\lambda_C = -\frac{1}{2C}\mathbf{w}_{\text{reg}}^{\mathrm{T}}\,\nabla E_{\text{in}}\,(\mathbf{w}_{\text{reg}})$$

$\mathbf{w}_{\text{reg}}$ minimizes $E_{\text{in}}(\mathbf{w}) + \lambda_C\mathbf{w}^{\mathrm{T}}\Gamma^{\mathrm{T}}\Gamma\mathbf{w}$. [Hint: use the previous part to solve for $\mathbf{w}_{\text{reg}}$ as an equality constrained optimization problem using the method of Lagrange multipliers.] (10pt)
(d) Show that the following hold for λ_C :
(i) If $\mathrm{w}_{\text{lin}}^{\mathrm{T}}\,\Gamma^{T}\Gamma\mathbf{w}_{\text{lin}} \leq C$ then $\lambda_C = 0$ ($\mathbf{w}_{\text{lin}}$ itself satisfies the constraint). (5pt)
(ii) If $\mathrm{w}_{\text{lin}}^{\mathrm{T}}\,\Gamma^{\mathrm{T}}\Gamma\mathrm{w}_{\text{lin}} > C$, then $\lambda_C > 0$ (the penalty term is positive). (5pt)
(iii) If $\mathrm{w}_{\text{lin}}^{\mathrm{T}}\,\Gamma^{\mathrm{T}}\Gamma\mathrm{w}_{\text{lin}} > C$, then λ_C is a strictly decreasing function of C. [Hint: show that $\frac{d\lambda_C}{dC} < 0$ for $C \in \left[0, \mathbf{w}_{\text{lin}}^{T}\,\Gamma^{T}\Gamma\mathbf{w}_{\text{lin}}\right]$]. (5pt)

Solution

(a) Since $\mathbf{w}_{\text{lin}}^{\text{T}} \Gamma^{\text{T}} \Gamma \mathrm{w}_{\text{lin}} \leq C$, which has already suitable for the constrains, so $\mathbf{w}_{\text{reg}} = \mathbf{w}_{\text{lin}}$.

(b) From the figure of contours, we can find that the minimum of E_{in} which also fit the constraint is on the boundary of the constraint, i.e. the intersection of the objective function's contour and the constraint. So

$$\mathbf{w}_{\text{reg}}^{\text{T}} \Gamma^{\text{T}} \Gamma \mathbf{w}_{\text{reg}} = C.$$

(The explanation of why the contours of constant E_{in} are the ellipsoids can be found in the textbook.)

(c) The original constrained problem is equivalent to solving the following unconstrained problem with Lagrange multipliers:

$$\min_{\mathbf{w}} L(\mathbf{w}, \lambda_C) = E_{\text{in}}(\mathbf{w}) + \lambda_C(\mathbf{w}^{\text{T}} \Gamma^{\text{T}} \Gamma \mathbf{w} - C)$$

And its gradients are:

$$\nabla_{\mathbf{w}} L(\mathbf{w}, \lambda_C) = \nabla E_{\text{in}}(\mathbf{w}) + 2\lambda_C \Gamma^{\text{T}} \Gamma \mathbf{w}$$

$$\frac{\partial}{\partial \lambda_{\mathbf{C}}} L(\mathbf{w}, \lambda_C) = \mathbf{w}^{\text{T}} \Gamma^{\text{T}} \Gamma \mathbf{w} - C$$

Since $\mathbf{w}_{reg}$ is a solution to the original constrained problem, it must also be a solution to the equivalent unconstrained problem, this means that

$$\nabla E_{\text{in}}(\mathbf{w}_{\text{reg}}) + 2\lambda_C \Gamma^{\text{T}} \Gamma \mathbf{w}_{\text{reg}} = 0$$

$$\mathbf{w}_{\text{reg}}^{\text{T}} \Gamma^{\text{T}} \Gamma \mathbf{w}_{\text{reg}} - C = 0$$

So we can get

$$\lambda_C = -\frac{1}{2C} \mathbf{w}_{\text{reg}}^{T} \nabla E_{\text{in}}(\mathbf{w}_{\text{reg}})$$

(d) [Note: also acceptable to solve by KKT]

(i) If $\mathrm{w}_{\text{lin}}^{\text{T}} \Gamma^{T} \Gamma \mathbf{w}_{\text{lin}} \leq C$, we know that $\mathbf{w}_{\text{reg}} = \mathbf{w}_{\text{lin}}$, and consequently $\nabla E_{\text{in}}(\mathbf{w}_{\text{reg}}) = 0$, which implies that $\lambda_C = 0$.

(ii) If $\mathrm{w}_{\text{lin}}^{\text{T}} \Gamma^{\text{T}} \Gamma \mathrm{w}_{\text{lin}} > C$, let us assume that $\lambda_C = 0$. This means that $\mathbf{w}_{\text{reg}}$ minimizes

$$E_{\text{in}}(\mathbf{w}) + \lambda_C(\mathbf{w}^{\text{T}} \Gamma^{\text{T}} \Gamma \mathbf{w} - C) = E_{\text{in}}(\mathbf{w}),$$

so we have $\mathbf{w}_{\text{reg}} = \mathbf{w}_{\text{lin}}$ and

$$\mathbf{w}_{\text{reg}}^{\text{T}} \Gamma^{\text{T}} \Gamma \mathbf{w}_{\text{reg}} = \mathbf{w}_{\text{lin}}^{\text{T}} \Gamma^{\text{T}} \Gamma \mathbf{w}_{\text{lin}} > C,$$

which is not possible since $\mathbf{w}_{\text{reg}}^{\text{T}} \Gamma^{\text{T}} \Gamma \mathbf{w}_{\text{reg}} \leq C$ by definition. Hence, we have $\lambda_C > 0$.

(iii) As $\mathbf{w}_{\text{lin}}^{\text{T}} \Gamma^{\text{T}} \Gamma \mathbf{w}_{\text{lin}} > C$, we have $\lambda_C > 0$ which means that $\mathbf{w}_{\text{reg}}^{T} \nabla E_{\text{in}}(\mathbf{w}_{\text{reg}}) < 0$. Now, if we compute the derivative relative to C, we get

$$\frac{d\lambda_C}{dC} = \frac{1}{2C^2} \mathbf{w}_{\text{reg}}^{T} \nabla E_{\text{in}}(\mathbf{w}_{\text{reg}}) < 0$$

Problem 2 [The Lasso algorithm] Rather than a soft order constraint on the squares of the weights, one could use the absolute values of the weights:

$$\min E_{\text{in}}(\mathbf{w})$$
$$\text{subject to } \sum_{i=0}^{d} |w_i| \leq C$$

The model is called the lasso algorithm.
(a) Formulate and implement this as a quadratic program. (10pt)
(b) What is the augmented error and discuss the algorithm for solving it. You can solve this problem using iterative soft-thresholding algorithm or a gradient projection method and present your pseudocode. (15pt)

Solution
(a) We can set $E_{\text{in}}(\mathbf{w})$ to be the L_2 loss, i.e.

$$E_{\text{in}}(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N}\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n - y_n\right)^2 = \frac{1}{N}\|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2$$

where $\mathbf{X}$ is the matrix of feature vectors, $\mathbf{y}$ is the vector of labels, and N is the number of data points, $\mathbf{x}_n$ is the feature vector of the n-th data point and y_n is the label.

For the objective function, we can rewrite it as:

$$\begin{aligned}\min_{\mathbf{w},\mathbf{t}} E_{\text{in}}(\mathbf{w}) &= \frac{1}{N}\|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2 \\ &= \frac{1}{N}(\mathbf{X}\mathbf{w} - \mathbf{y})^T(\mathbf{X}\mathbf{w} - \mathbf{y}) \\ &= \frac{1}{N}(\mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w} - 2\mathbf{w}^T\mathbf{X}^T\mathbf{y} + \mathbf{y}^T\mathbf{y})\end{aligned}$$

which is a quadratic function of $\mathbf{w}, \mathbf{t}$, and the definition of $\mathbf{t}$ is as followed. (5pt)
For the constrains, we can slack the variables by letting $|w_i| \leq t_i$ for $i = 1, 2, \cdots, d$, and $t_i \geq 0$. Then we can rewrite the constrain as:

$$\begin{aligned}\sum_{i=0}^{d} t_i &\leq C \\ t_i &\geq 0, \qquad i = 1, 2, \cdots, d \\ -t_i \leq w_i &\leq t_i, \qquad i = 1, 2, \cdots, d\end{aligned}$$

which are linear constraints of $\mathbf{w}, \mathbf{t}$.
Since the optimization problem is a quadratic programming problem with linear constrains, so it is a quadratic programming. (5pt)

(b) The augmented error is defined as:

$$E_{\text{aug}}(\mathbf{w}) = E_{\text{in}}(\mathbf{w}) + \lambda\sum_{i=0}^{d}|w_i| = E_{\text{in}}(\mathbf{w}) + \lambda\|\mathbf{w}\|_1$$

where λ and is the hyperparameter. (5pt)
Since L_1 norm is not differentiable, so we cannot simply use the gradient methods, but we can use the iterative soft-thresholding algorithm to solve it.

Define the regularization term to be $h(\mathbf{x}) = \|\mathbf{x}\|_1$.
The L_1 regularization's proximal term is $\text{prox}_{\lambda h}(\mathbf{x}) = \arg\min_{\mathbf{z}}\left\{\frac{1}{2}\|\mathbf{z} - \mathbf{x}\|_2^2 + \lambda h(\mathbf{z})\right\}$.
Since the proximal term is seperatable, so we can decompose into item by item optimization with soft-thresholding.
i.e.

$$(\text{prox}_{\lambda h}(\mathbf{x}))_i = \psi_{\text{st}}(x_i, \lambda)$$

where ψ_{st} is the soft-thresholding function.
Then we analyze the soft-thresholding function:

$$\psi_{\text{st}}(x, \lambda) = \arg\min_{z_i} \left\{ \frac{1}{2}(z_i - x_i)^2 + \lambda |z_i| \right\}$$

- If $z_i \geq 0$, then $\arg\min_{z_i} \left\{ \frac{1}{2} z_i^2 + (\lambda - x_i) z_i + \frac{1}{2} x_i^2 \right\}$, which is a simple quadratic function.
  1. If $x_i \geq \lambda$, then $z_i = x_i - \lambda \geq 0$
  2. If $x_i < \lambda$, then $z_i = 0$

- If $z_i < 0$, then $\arg\min_{z_i} \left\{ \frac{1}{2} z_i^2 - (\lambda + x_i) z_i + \frac{1}{2} x_i^2 \right\}$, which is also a simple quadratic function.
  1. If $x_i \leq -\lambda$, then $z_i = x_i + \lambda \leq 0$
  2. If $x_i > -\lambda$, then $z_i = 0$

So combine all these cases together, we can get the soft-thresholding function:

$$\psi_{\text{st}}(x_i, \lambda) = \begin{cases} x_i - \lambda, & x_i > \lambda \\ 0, & |x_i| \leq \lambda \\ x_i + \lambda, & x_i < -\lambda \end{cases}$$

So with the soft-thresholding function, we can solve the Lasso problem by applying the proximal gradient method.

The pseudocode is shown in Algorithm 1.

Algorithm 1 Proximal Gradient Method for Lasso Problem

1: **for** $t = 0, 1, 2, \cdots$ **do**
2: $\quad \mathbf{w}^{(t+1)} \leftarrow \text{prox}_{\eta_t \lambda h}(\mathbf{w}^{(t)} - \eta_t \nabla E_{\text{in}}(\mathbf{w}^{(t)}))$
3: **end for**

[Note: Other equivalent forms of the iterative soft-thresholding algorithm are also acceptable. It's also acceptable to combine the algorithm into the pseudocode provided the students stated clearly.] (10pt)

Problem 3 Similar to problem 3 in assignment 3 and assignment 4, you need to use the SUV dataset to implement (using Python or MATLAB) the L_1/L_2 regularization (penalty/augmented).

- Present your code. (15pt)
- Present the path plots of L_1 and L_2 regularization. (Notice: you need to mark the selected value of the regular parameter) (10pt)
- Analyze the weight difference between L_1 and L_2 regularization. (Notice: you need to describe the similarities and differences between the solutions of path plots) (10pt)
- If you only want to build a model that contains 2 variables, which two features would you choose? (5pt)

Solution

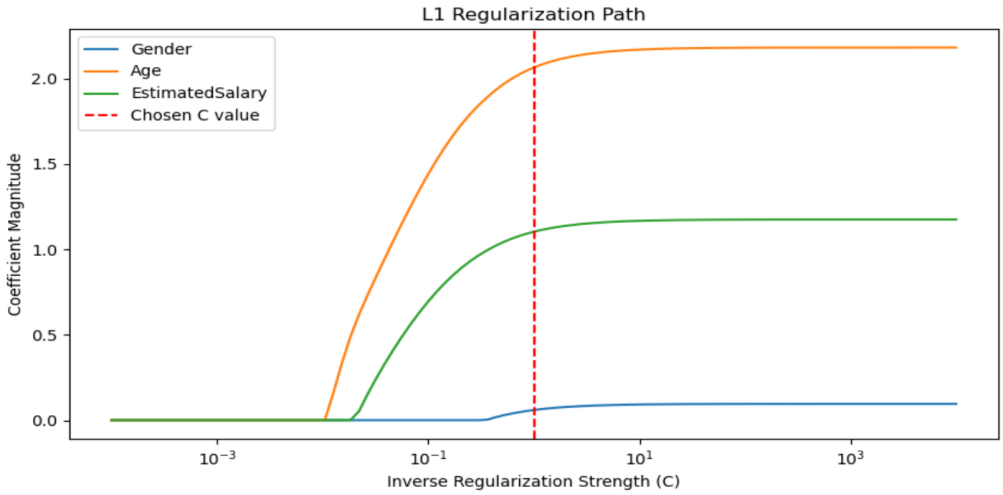


Figure 1: Path plot of L_1 regularization

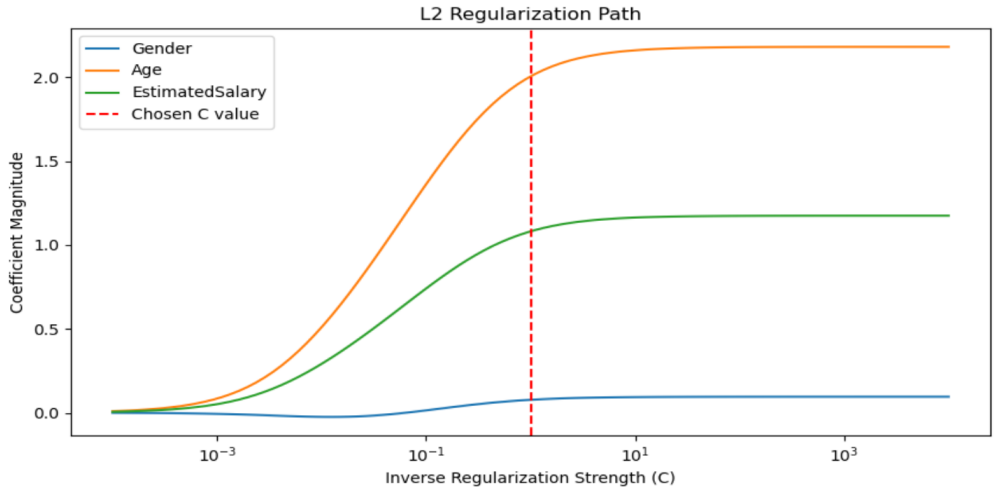


Figure 2: Path plot of L_2 regularization

Students should provide codes, otherwise they will get 0 pt for the whole Problem 3.

(a) The key point is to check whether students add correct regularization terms to the loss.

(b) The example path plots of L_1 and L_2 regularization are shown in Figure 1 and Figure 2. [Note: Students should plot paths of both kinds of regularization (each picture 5 pt). In each picture, they should draw the paths of at least 3 features (Age, Gender, and EstimatedSalary, each 1 pt), and mark the selected value of the regular parameter (1 pt). Most importantly, coefficients should shrink towards 0 as λ **increases**, otherwise will be subtracted half of all points.]

(c) Both L_1 and L_2 regularization has smaller weights as the regularization parameter λ increases. (4pt) The difference is that L_1 regularization can make some weights to be zero, which means that L_1 regularization can do feature selection. (3pt) In contrast, L_2 regularization can only make the weights smaller but not zero (or shrinks the paths towards zero more smoothly). (3pt)

(d) "Age" and "EstimatedSalary", since they drop to 0 slower than other features as λ Increases (or they have larger magnitudes than other features), indicating that these predictors are more important for the model.