Introduction to Machine Learning
Fall 2025
University of Science and Technology of China

Lecturer: Jie Wang | Homework 5
Posted: Nov. 29, 2025 | Due: Dec. 10, 2025

Notice, to get the full credits, please show your solutions step by step.

Exercise 1: Proximal Operator

For a convex function $f : \mathbb{R}^n \to \mathbb{R}$, we define its proximal operator at $\mathbf{x}$ by

$$\text{prox}_f(\mathbf{x}) = \underset{\mathbf{u}\in\text{dom}\, f}{\arg\min} \left\{ f(\mathbf{u}) + \frac{1}{2}\|\mathbf{u}-\mathbf{x}\|^2 \right\}.$$

1. The proximal operator has the following properties.

   (a) If f is proper and close (which means $\mathbf{epi}f$ is close), please show that for any $\mathbf{x} \in \mathbb{R}^n$, $\text{prox}_f(\mathbf{x})$ exists and is unique. You can use the properties we have proved in Homework 4 directly.

   (b) If f is proper and close, then show that $\mathbf{u} = \text{prox}_f(\mathbf{x})$ if and only if $\mathbf{x}-\mathbf{u} \in \partial f(\mathbf{u})$.

   (c) (Optional) Please show that if $\mathbf{u} = \text{prox}_f(\mathbf{x}), \mathbf{v} = \text{prox}_f(\mathbf{y})$, then

   $$\langle \mathbf{u}-\mathbf{v}, \mathbf{x}-\mathbf{y}\rangle \geq \|\mathbf{u}-\mathbf{v}\|_2^2,$$

   which means prox_f is firmly nonexpansive. Then show that this implies nonexpansive

   $$\left\|\text{prox}_f(\mathbf{x}) - \text{prox}_f(\mathbf{y})\right\|_2 \leq \|\mathbf{x}-\mathbf{y}\|_2.$$

2. Please show that the proximal operator satisfies the following equations.

   (a) For $\lambda \neq 0$ and $a \in \mathbb{R}^n$, we let $h(\mathbf{x}) = f(\lambda\mathbf{x}+\mathbf{a})$, then $\text{prox}_h(\mathbf{x}) = \frac{1}{\lambda}\left(\text{prox}_{\lambda^2 f}(\lambda\mathbf{x}+\mathbf{a}) - \mathbf{a}\right)$.

   (b) For $\lambda > 0$, we let $h(\mathbf{x}) = \lambda f\left(\frac{\mathbf{x}}{\lambda}\right)$, then $\text{prox}_h(\mathbf{x}) = \lambda\,\text{prox}_{\lambda^{-1}f}\left(\frac{\mathbf{x}}{\lambda}\right)$.

   (c) For $\mathbf{a} \in \mathbb{R}^n$, we let $h(\mathbf{x}) = f(\mathbf{x}) + \mathbf{a}^\top\mathbf{x}$, then $\text{prox}_h(\mathbf{x}) = \text{prox}_f(\mathbf{x}-\mathbf{a})$.

3. Please find the proximal operator of the following functions.

   (a) $f(\mathbf{x}) = \|\mathbf{x}\|_2$

   (b) $f(\mathbf{x}) = I_C(\mathbf{x})$, where C is a convex set.

4. Consider the convex optimization problem

   $$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) + \tilde{I}_D(\mathbf{x}), \tag{1}$$

   where $D \subseteq \mathbb{R}^n$ is a closed convex set and $\tilde{I}_D(\mathbf{x})$ is the extended-value extension of its indicator function $I_D(\mathbf{x})$.

   (a) Write down the optimality condition and the proximal operator of Problem (1).

(b) Find the relationship between (1) and the constrained optimization problem

$$\min_{\mathbf{x}\in D} f(\mathbf{x}).$$

5. Recall the convex optimization problem in Lecture 08.

$$\min_{\mathbf{x}\in\mathbb{R}^n} F(\mathbf{x}).$$

Please rewrite $p(\mathbf{x}_c)$ using proximal operator.

6. If we use ISTA to solve the following problems, please find the $p(\mathbf{w})$ of them.

(a) The Elastic Net optimization problem, defined as:

$$\min_{\mathbf{w}\in\mathbb{R}^n} \frac{1}{n}\|\mathbf{y}-\mathbf{X}\mathbf{w}\|_2^2 + \lambda_1\|\mathbf{w}\|_1 + \lambda_2\|\mathbf{w}\|_2^2,$$

where $\lambda_1 > 0$, and $\lambda_2 > 0$.

(b) The Group Lasso optimization problem, defined as:

$$\min_{\mathbf{w}\in\mathbb{R}^n} \frac{1}{n}\|\mathbf{y}-\mathbf{X}\mathbf{w}\|_2^2 + \lambda\sum_{j=1}^{G}\|\mathbf{w}_{\mathcal{G}_j}\|_2,$$

where $\lambda > 0$.

Solution: ■

Exercise 2: Proximal Gradient

Consider the following convex optimization problem

$$\begin{aligned} &\min_{\mathbf{x}} F(\mathbf{x}) \\ &\text{s.t.} \mathbf{x} \in D \end{aligned} \tag{2}$$

where $F : \mathbb{R}^n \to \overline{\mathbb{R}}$ is a proper convex function and $D \subseteq \mathbb{R}^n$ is a nonempty convex set with $D \subseteq \mathbf{dom}\ F$. Suppose that the problem (2) is solvable, and **we do not require the differentiability of** F.

1. If $\mathbf{x} \in \mathbf{int}\,(\mathbf{dom}\ F) \cap D$ and there exists a $\mathbf{g} \in \partial F(\mathbf{x})$ such that

   $$\langle \mathbf{g}, \mathbf{y} - \mathbf{x} \rangle \geq 0, \forall\, \mathbf{y} \in D,$$

   show that $\mathbf{x}$ is optimal.

2. Please give an example to show that $\partial F(\mathbf{x})$ can be empty.

3. Suppose $f : \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable, and the Hessian matrix of f is $\mathbf{H}(\mathbf{x})$.

   (a) Let $\lambda_{\max}(\mathbf{x})$ represents the largest eigenvalue of $\mathbf{H}(\mathbf{x})$. If

   $$\lambda_{\max}(\mathbf{x}) \leq L, \quad \forall \mathbf{x} \in \mathbb{R}^n,$$

   please show that:

   $$f(\mathbf{y}) \leq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{L}{2}\|\mathbf{y} - \mathbf{x}\|^2.$$

   (b) Let $\lambda_{\min}(\mathbf{x})$ represent the smallest eigenvalue of $\mathbf{H}(\mathbf{x})$. If f is strongly convex with convexity parameter $\mu > 0$, please show that:

   $$\mu \leq \lambda_{\min}(\mathbf{x}), \quad \forall \mathbf{x} \in \mathbb{R}^n.$$

In many cases, the function F can be decomposed into $F = f + g$, where $g : \mathbb{R}^n \to \overline{\mathbb{R}}$ is a continuous convex function, and $f : \mathbb{R}^n \to \mathbb{R}$ is a convex and continuously differentiable function, whose gradient is Lipschitz continuous with the constant L. We can use ISTA, which has been introduced in Lecture 08, to find $\min\limits_{\mathbf{x} \in \mathbb{R}^n} F(\mathbf{x})$.

4. For a **given** point $\mathbf{x}_c$, we consider the following quadratic approximation of F:

   $$Q(\mathbf{x}; \mathbf{x}_c) = f(\mathbf{x}_c) + \langle \nabla f(\mathbf{x}_c), \mathbf{x} - \mathbf{x}_c \rangle + \frac{L}{2}\|\mathbf{x} - \mathbf{x}_c\|^2 + g(\mathbf{x}).$$

   Please show that it always admits a unique minimizer

   $$p(\mathbf{x}_c) = \mathbf{argmin}_{\mathbf{x} \in \mathbb{R}^n} Q(\mathbf{x}; \mathbf{x}_c).$$

5. If we use ISTA to solve the Lasso problem, show that

$$w_i^+ = \begin{cases} z_i + \dfrac{\lambda}{L}, & \text{if } z_i < -\dfrac{\lambda}{L}, \\ 0, & \text{if } |z_i| \le \dfrac{\lambda}{L}, \\ z_i - \dfrac{\lambda}{L}, & \text{if } z_i > \dfrac{\lambda}{L}, \end{cases}$$

where $\mathbf{z} = \mathbf{w}_k - \dfrac{2}{Ln}\mathbf{X}^\top(\mathbf{X}\mathbf{w}_k - \mathbf{y})$.

Solution: ■

Exercise 3: [1] ISTA with Backtracking

Suppose that we would like to apply ISTA to solve the convex optimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} F(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{x}), \tag{3}$$

where $g : \mathbb{R}^n \to \overline{\mathbb{R}}$ is a continuous convex function, and $f : \mathbb{R}^n \to \mathbb{R}$ is a convex and continuously differentiable function, whose gradient is Lipschitz continuous with the constant L. We assume that Problem (3) is solvable, i.e., there exists $\mathbf{x}^*$ such that

$$F(\mathbf{x}^*) = F^* = \min_{\mathbf{x}\in\mathbb{R}^n} F(\mathbf{x}).$$

In practice, however, a possible drawback of ISTA is that the Lipschitz constant L is not always known or computable. For instance, if $f(\mathbf{x}) = \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2$, the Lipschitz constant for ∇f depends on $\lambda_{\max}(\mathbf{A}^\top\mathbf{A})$, which is not always easily computable for large-scale problems. To tackle this problem, we always equip ISTA with the backtracking stepsize rule as shown in Algorithm 1.

Note that in Algorithm 1, Q_L and p_L are defined as

$$Q_L(\mathbf{x};\mathbf{x}_c) = f(\mathbf{x}_c) + \langle \nabla f(\mathbf{x}_c), \mathbf{x}-\mathbf{x}_c\rangle + \frac{L}{2}\|\mathbf{x}-\mathbf{x}_c\|_2^2 + g(\mathbf{x})$$
$$p_L(\mathbf{x}_c) = \operatorname*{\mathbf{argmin}}_{\mathbf{x}\in\mathbb{R}^n} Q_L(\mathbf{x};\mathbf{x}_c).$$

Algorithm 1 ISTA with Backtracking

1: **Input:** An initial point $\mathbf{x}_0$, an initial constant $L_0 > 0$, a threshold $\eta > 1$, and $k = 1$.
2: **while** the *termination condition* does not hold **do**
3: Find the smallest non-negative integer i_k such that with $\tilde{L} = \eta^{i_k} L_{k-1}$

$$F(p_{\tilde{L}}(\mathbf{x}_{k-1})) \le Q_{\tilde{L}}(p_{\tilde{L}}(\mathbf{x}_{k-1});\mathbf{x}_{k-1}). \tag{4}$$

4: $L_k \leftarrow \eta^{i_k} L_{k-1}$, $\mathbf{x}_k \leftarrow p_{L_k}(\mathbf{x}_{k-1})$,
5: $k \leftarrow k+1$,
6: **end while**

1. Show that the sequence $\{F(\mathbf{x}_k)\}$ produced by Algorithm 1 is non-increasing.

2. Show that Inequality (4) is satisfied for any $\tilde{L} \ge L$, where L is the Lipschitz constant of ∇f, thus showing that for Algorithm 1 one has $L_k \le \eta L$ for every $k \ge 1$.

3. Let $\{\mathbf{x}_k\}$ be the sequence generated by Algorithm 1. Show that for any $k \ge 1$ we have

$$F(\mathbf{x}_k) - F(\mathbf{x}^*) \le \frac{\eta L\|\mathbf{x}_0 - \mathbf{x}^*\|_2^2}{2k}, \ \forall \mathbf{x}^* \in \operatorname*{\mathbf{argmin}}_{\mathbf{x}\in\mathbb{R}^n} F(\mathbf{x}).$$

The above result means that the number of iterations of Algorithm 1 required to obtain an ε-optimal solution, i.e., an $\hat{\mathbf{x}}$ such that $F(\hat{\mathbf{x}}) - F(\mathbf{x}^*) \leq \varepsilon$, is at most

$$\left\lceil \frac{\eta L \|\mathbf{x}_0 - \mathbf{x}^*\|_2^2}{2\varepsilon} \right\rceil.$$

Solution: ■

Exercise 4: Programming Exercise: Naive Bayes Classifier

We provide you with a data set that contains spam and non-spam emails ("hw5_nb.zip"). Please use the Naive Bayes Classifier to detect the spam emails. Finish the following exercises by programming. You can use your favorite programming language.

1. Remove all the tokens that contain non-alphabetic characters.
2. Train the Naive Bayes Classifier on the training set according to Algorithm 2.
3. Test the Naive Bayes Classifier on the test set according to Algorithm 3. You may encounter a problem that the likelihood probabilities you calculate approach 0. How do you deal with this problem?
4. Compute the confusion matrix, accuracy, precision, recall, and F-score.
5. Without the Laplace smoothing technique, complete the steps again.

Algorithm 2 Training Naive Bayes Classifier

Input: The training set with the labels $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}$.

$\mathcal{V} \leftarrow$ the set of distinct words and other tokens found in $\mathcal{D}$
for each target value c in the labels set $\mathcal{C}$ **do**
  $\mathcal{D}_c \leftarrow$ the training samples whose labels are c
  $P(c) \leftarrow \frac{|\mathcal{D}_c|}{|\mathcal{D}|}$
  $T_c \leftarrow$ a single document by concatenating all training samples in $\mathcal{D}_c$
  $n_c \leftarrow |T_c|$
  for each word w_k in the vocabulary $\mathcal{V}$ **do**
    $n_{c,k} \leftarrow$ the number of times the word w_k occurs in T_c
    $P(w_k|c) = \frac{n_{c,k}+1}{n_c+|\mathcal{V}|}$
  end for
end for

Algorithm 3 Testing Naive Bayes Classifier

Input: An email $\mathbf{x}$. Let x_i be the i^{th} token in $\mathbf{x}$. $\mathcal{I} = \emptyset$.

for $i = 1, \ldots, |\mathbf{x}|$ **do**
  if $\exists\, w_{k_i} \in \mathcal{V}$ such that $w_{k_i} = x_i$ **then**
    $\mathcal{I} \leftarrow \mathcal{I} \cup i$
  end if
end for
predict the label of $\mathbf{x}$ by

$$\hat{y} = \arg\max_{c \in \mathcal{C}} P(c) \prod_{i \in \mathcal{I}} P(w_{k_i}|c)$$

Solution: ■

Exercise 5: Logistic Regression and Newton's Method

Given the training data $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{0, 1\}$. Let

$$\mathcal{I}^+ = \{i : i \in [n], y_i = 1\},$$
$$\mathcal{I}^- = \{i : i \in [n], y_i = 0\},$$

where $[n] = \{1, 2, \ldots, n\}$. We assume that $\mathcal{I}^+$ and $\mathcal{I}^-$ are not empty.
Then, we can formulate the logistic regression of the form.

$$\min_{\mathbf{w}} \; L(\mathbf{w}) = -\frac{1}{n}\sum_{i=1}^{n}\left(y_i \log\left(\frac{\exp(\langle \mathbf{w}, \overline{\mathbf{x}}_i\rangle)}{1+\exp(\langle \mathbf{w}, \overline{\mathbf{x}}_i\rangle)}\right) + (1-y_i)\log\left(\frac{1}{1+\exp(\langle \mathbf{w}, \overline{\mathbf{x}}_i\rangle)}\right)\right), \quad (5)$$

where $\mathbf{w} \in \mathbb{R}^{d+1}$ is the model parameter to be estimated and $\overline{\mathbf{x}}_i^\top = (1, \mathbf{x}_i^\top)$.

1. (a) Suppose that the training data is strictly linearly separable, that is, there exists $\hat{\mathbf{w}} \in \mathbb{R}^{d+1}$ such that

$$\langle \hat{\mathbf{w}}, \overline{\mathbf{x}}_i\rangle > 0, \, \forall \, i \in \mathcal{I}^+,$$
$$\langle \hat{\mathbf{w}}, \overline{\mathbf{x}}_i\rangle < 0, \, \forall \, i \in \mathcal{I}^-.$$

   Show that problem (5) has no solution.

   (b) Suppose that the training data is NOT linearly separable, that is, for all $\mathbf{w} \in \mathbb{R}^{d+1}$, there exists $i \in [n]$ such that

$$\langle \mathbf{w}, \overline{\mathbf{x}}_i\rangle < 0, \text{ if } i \in \mathcal{I}^+,$$

   or

$$\langle \mathbf{w}, \overline{\mathbf{x}}_i\rangle > 0, \text{ if } i \in \mathcal{I}^-.$$

   Show that problem (5) always admits a solution.

2. Suppose that $\overline{\mathbf{X}} = (\overline{\mathbf{x}}_1, \overline{\mathbf{x}}_2, \ldots, \overline{\mathbf{x}}_n)^\top \in \mathbb{R}^{n\times(d+1)}$ and $\mathbf{rank}(\overline{\mathbf{X}}) = d + 1$. Show that $L(\mathbf{w})$ is strictly convex, i.e., for all $\mathbf{w}_1 \neq \mathbf{w}_2$,

$$L(t\mathbf{w}_1 + (1-t)\mathbf{w}_2) < tL(\mathbf{w}_1) + (1-t)L(\mathbf{w}_2), \forall \, t \in (0, 1).$$

Solution: ■

Exercise 6: Convergence of Stochastic Gradient Descent for Convex Function

Consider an optimization problem

$$\min_{\mathbf{w}} F(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{w}), \tag{6}$$

where the objective function F is continuously differentiable and strongly convex with convexity parameter $\mu > 0$. Suppose that the gradient of F, i.e., ∇F, is Lipschitz continuous with Lipschitz constant L, and F can attain it minimum F^* at $\mathbf{w}^*$. We use the stochastic gradient descent(SGD) algorithm introduced in Lecture 12 to solve the problem (6). Let the solution sequence generated by SGD be $(\mathbf{w}_k)$.

1. Please show that $\forall \mathbf{w} \in \mathbf{dom}\ F$, the following inequality

$$F(\mathbf{w}) - F^* \leq \frac{1}{2\mu}\|\nabla F(\mathbf{w})\|^2 \tag{7}$$

   holds, and interpret the role of strong convexity based on this.

2. Recall that with a fixed stepsize $\alpha \in [0, \frac{1}{LM_G}]$ where M_G (as well as the following M) is a parameter regarding the upper bound of the variance of stochastic gradient in SGD, the sequence $(\mathbb{E}[F(\mathbf{w}_k)])$ generated by SGD converges to a neighborhood of F^* with a linear rate, i.e,

$$\mathbb{E}_{\xi_0:\xi_{k-1}}[F(\mathbf{w}_k) - F^*] \leq \frac{LM}{2\mu}\alpha + (1-\mu\alpha)^k(F(\mathbf{w}_0) - F^* - \frac{LM}{2\mu}\alpha) \xrightarrow{\text{linear}} \frac{LM}{2\mu}\alpha.$$

   (a) In practice, for the same problem, SGD enjoys less time cost but more iteration steps than gradient descent methods and may suffer from non-convergence. As a trade-off between SGD and gradent descent approaches, consider using mini-batch samples to estimate the full gradient. Taking k^{th} iteration as an example, instead of picking a single sample, we randomly select a subset $\mathcal{S}_k$ of the sample indices to compute the update direction

$$\mathbf{g}_k(\xi_k) = \frac{1}{|\mathcal{S}_k|}\sum_{i\in\mathcal{S}_k} \nabla f_i(\mathbf{w}_k)$$

   where ξ_k is the selected samples. For simplicity, suppose that the mini-batches in all iterations are of constant size, i.e., $|\mathcal{S}_k| = n_m$, and the stepsize α is fixed. Please show that for mini-batch SGD, there holds

$$\mathbb{E}_{\xi_0:\xi_{k-1}}[F(\mathbf{w}_k) - F^*] \leq \frac{LM}{2\mu n_m}\alpha + (1-\mu\alpha)^k(F(\mathbf{w}_0) - F^* - \frac{LM}{2\mu n_m}\alpha) \xrightarrow{\text{linear}} \frac{LM}{2\mu n_m}\alpha.$$

   Moreover, point out the advantage of mini-batch SGD compared to SGD in terms of the number of the iteration step.

   (b) The expected optimality gap of SGD, i.e., $\mathbb{E}_{\xi_0:\xi_{k-1}}[F(\mathbf{w}_k) - F^*]$, fails to converge to zero. In order to alleviate this problem, we consider a strategy of diminishing stepsize α_k. Suppose that the SGD method is run with a stepsize sequence (α_k)

such that, for all $k \in \mathbb{N}$,$\alpha_k = \frac{\beta}{\gamma+k}$ for some $\beta > \frac{1}{\mu}$ and $\gamma > 0$ satisfying $\alpha_0 \leq \frac{1}{LM_G}$. Please show that $\forall k \in \mathbb{N}$, we have

$$\mathbb{E}_{\xi_0:\xi_{k-1}}[F(\mathbf{w}_k) - F^*] \leq \frac{\tau}{\gamma + k},$$

where $\tau = \max\{\frac{\beta^2 LM}{2(\beta\mu-1)}, \gamma(F(\mathbf{w}_0) - F^*)\}$.

Solution: ■

Exercise 7: Programming Exercise: Logistic Regression

We provide you with a dataset of handwritten digits called MNIST[1], that contains a training set of 60000 examples and a test set of 10000 examples ("hw5_lr.zip"). Each image in this dataset has 28×28 pixels and the associated label is the handwritten digit—that is, an integer from the set $\{0, 1, \cdots, 9\}$—in the image. In this exercise, you need to build a logistic regression classifier to *predict if a given image has the handwritten digit* 6 *in it or not*. You can use your favorite programming language to finish this exercise.

1. Normalize the data matrix and please find a Lipschitz constant of $\nabla L(\mathbf{w})$, where $L(\mathbf{w})$ is the objective function of the logistic regression after normalizing and $\mathbf{w}$ is the model parameter to be estimated.

2. (a) Use the gradient descent algorithm (GD), which is a special case of ISTA introduced in Lecture 09, and SGD to train the logistic regression classifier on the training set, respectively. Evaluate the classification accuracy on the training set after each iteration. Stop the iteration when Accuracy $\geq 95\%$ or total steps are more than 2000. Please plot the accuracy of these two classifiers (the one trained by GD and the other trained by SGD) versus the iteration step on one graph.

   (b) Compare the total iteration counts and the total time cost of the two methods (GD and SGD), respectively. Please report your result.

   (c) Compare the confusion matrix, precision, recall and F1 score of the two classifiers (the one trained by GD and the other trained by SGD). Please report your result.

3. (a) The order of data samples affects the convergence of SGD. Implement and compare the following sampling strategies for training the logistic regression classifier(other experimental setup details is in line with 2.(a)):

   - Random Sampling without Replacement: At the start of each epoch, shuffle the entire training set and perform parameter updates by iterating through all samples sequentially.
   - Random Sampling with Replacement: At each iteration, uniformly sample a single data point from the full training set to update parameters.
   - Mini-batch Sampling: In each epoch, partition the training set into fixed-size sequential mini-batches and perform updates iteratively.

   (b) Please plot the accuracy of these classifiers versus the iteration step on one graph. Compare the convergence speeds and stability of these classifiers. Please report your result.

Solution: ■

[1]It was created by the National Institute of Standards and Technology (NIST).

References

[1] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.