Homework 2 40pt (Math 466, Spring 2025)

Due Date: Feb. 27, Thur, 10 pm

Instructions

- Submissions must be made in PDF format. If there are coding problems, please provide source code that directly generates the reported results.
- In written problems, if no derivation is asked for, just give the answer. Otherwise, try to give the derivation in a clear and mathematically rigorous way.
- For coding problems, please include a mini-report summarizing your experiment, along with any relevant figures or tables.

  When creating plots or tables, ensure they are clear and easy to interpret by using different colors or markers, and by adding axis labels, titles, captions, etc. to enhance readability.

  You are also required to submit the code used to generate the results presented in your mini-report. This can be provided as either a .ipynb or .py file.
- Collaboration and AI tools are allowed, but you must write your solutions independently and acknowledge collaborators/AI tools.

Questions

1. **(6 pt) This problem is about the existence and uniqueness of an optimization problem.**

   Optimal transport studies how to move a pile of earth to a target hole with least effort. It induces a distance on probability space, which is called Wasserstein distance. Both optimal transport itself and Wasserstein distance play important roles in many machine learning tasks.

   In this problem, we look at optimal transport between two discrete distributions

$$\alpha = \sum_{i=1}^{n} a_i \delta_{\mathbf{x}_i}, \quad \beta = \sum_{j=1}^{m} b_j \delta_{\mathbf{y}_j},$$

where $\mathbf{x}_i, \mathbf{y}_j \in \mathbb{R}^d$, $a_i \geq 0, b_j \geq 0$ and $\sum_{i=1}^n a_i = 1, \sum_{j=1}^m b_j = 1$. Denote

$$\mathbf{a} = (a_1, \cdots, a_n)^\top \in \mathbb{R}^n, \mathbf{b} = (b_1, \cdots, b_n)^\top \in \mathbb{R}^m.$$

Let $C_{ij} \in \mathbb{R}$ be the cost of moving unit mass from $\mathbf{x}_i$ to $\mathbf{y}_j$, and denote

$$C = (C_{ij})_{ij} \in \mathbb{R}^{n \times m}.$$

(a) (3pt) Monge Problem: consider all maps from $\{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ to $\{\mathbf{y}_1, \cdots, \mathbf{y}_m\}$ that move the mass from α to β exactly, the Monge problem seeks the map that gives the minimal cost. To be precise, let

$$\mathcal{T} := \left\{ T : \{1, \cdots, n\} \to \{1, \cdots, m\} \mid b_j = \sum_{i:T(i)=j} a_i \right\}.$$

The Monge problem is

$$\min_{T \in \mathcal{T}} \sum_{i=1}^n C_{i,T(i)}.$$

- Does the Monge problem always admit at least one minimizer? If yes, provide your reasoning (there is no need to prove it). Otherwise, give a counter-example.
- If the Monge problem admits at least one minimizer, is the minimizer always unique? If yes, provide your reasoning (there is no need to prove it). Otherwise, give a counter-example.

(b) (3pt) Kantorovich problem seeks the coupling between α, β that gives the minimal cost. To be precise, let

$$U(\mathbf{a}, \mathbf{b}) := \left\{ P \in \mathbb{R}_+^{n \times m} \mid \sum_{j=1}^m P_{ij} = a_i, i = 1, \cdots, n, \sum_{i=1}^n P_{ij} = b_j, j = 1, \cdots, m \right\}.$$

The Kantorovich problem is

$$\min_{P \in U(\mathbf{a}, \mathbf{b})} \sum_{i=1}^n \sum_{j=1}^m C_{ij} P_{ij}.$$

- Does the Kantorovich problem always admit at least one minimizer? If yes, provide your reasoning (there is no need to prove it). Otherwise, give a counter-example.
- If the Kantorovich problem admits at least one minimizer, is the minimizer always unique? If yes, provide your reasoning (there is no need to prove it). Otherwise, give a counter-example.

2. **(14 pt) This problem is about the convexity.**

(a) (5pt) Please choose one of the problems, logistic regression or support vector machine, to answer. If you answer both, your grade will be based on the higher score.

Let $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ be a training dataset with $\mathbf{x}_i \in \mathbb{R}^d$ being feature vectors and $y_i \in \{-1, 1\}$. Assume that $\mathbf{x}_i$ are centralized.

- Recall the logistic regression problem has the following formulation

$$f(\mathbf{w}) = \sum_{i=1}^{N} \log(1 + \exp(-y_i \mathbf{w}^\top \mathbf{x}_i)).$$

Prove that f is convex in $\mathbf{w}$.

- Recall the soft-margin SVM problem has the following formulation

$$f(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \max\{0, 1 - y_i \mathbf{w}^\top \mathbf{x}_i\} + \frac{\lambda}{2}\|\mathbf{w}\|_2^2,$$

where $\lambda > 0$. Prove that f is convex in $\mathbf{w}$.

(b) (9pt) Let

$$\Sigma_n := \{\mathbf{a} \in \mathbb{R}^n \mid a_i \geq 0, i = 1, \cdots, n, \sum_{i=1}^{n} a_i = 1\},$$

and consider $\mathbf{a}, \mathbf{b} \in \Sigma_n$. $\mathbf{a}, \mathbf{b}$ can be viewed as two discrete probability distributions with the same support. The KL divergence is a measure of how one probability distribution is different from another reference distribution. It is not a norm but it is widely used in machine learning. The KL divergence of $\mathbf{a}, \mathbf{b}$ is defined as

$$KL(\mathbf{a}||\mathbf{b}) := \sum_{i=1}^{n} a_i \log\left(\frac{a_i}{b_i}\right),$$

with the convention $0/0 = 0, 0\log 0 = 0$.

(b.1) Prove that $f : \mathbb{R}_{++} \to \mathbb{R}, x \mapsto x \log x$ is convex.

(b.2) Prove a special case of Jensen's inequality: for a convex function $\varphi : \mathbb{R}_{++} \to \mathbb{R}$, if $x_1, x_2, \cdots, x_n > 0$, the weights $p_1, \cdots, p_n \geq 0$ and $\sum_{i=1}^n p_i = 1$, then

$$\varphi\left(\sum_{i=1}^{n} p_i x_i\right) \leq \sum_{i=1}^{n} p_i \varphi(x_i).$$

(b.3) Prove that $KL(\mathbf{a}||\mathbf{b}) \geq 0$.

3. **(20 pt) This problem is about optimization algorithm.**

Recap 1. LASSO

Consider a training dataset $\{(x_i, y_i)\}_{i=1}^n$, where $x_i \in \mathbb{R}^d, y_i \in \mathbb{R}$. Assume that the inputs are centralized. LASSO aims to solve the following minimization problem for the weight vector w. Here λ is the regularization parameter.

$$\min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^n (x_i^\top w - y_i)^2 + \lambda \|w\|_1. \tag{1}$$

Recap 2. Proximal gradient

Consider an optimization problem

$$\min_{w \in \mathbb{R}^d} f(w) + r(w) \tag{2}$$

where f is differentiable and r is convex but may not be differentiable. The update of the proximal gradient is

$$w^{(k+1)} = \mathrm{prox}_{\eta r} \left(w^{(k)} - \eta \nabla f(w^{(k)}) \right). \tag{3}$$

Here η is the step-size (learning rate). If f is L-smooth, we can take $\eta = \frac{1}{L}$.

Recap 3. Stochastic gradient

Consider the optimization problem

$$\min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^n f_i(w), \tag{4}$$

where each function $f_i(w)$ corresponds to the loss associated with a data sample or mini-batch. In large-scale machine learning problems, computing the full gradient $\nabla F(w) = \frac{1}{n} \sum_{i=1}^n \nabla f_i(w)$ at each iteration can be computationally expensive. The Stochastic Gradient Method (SGM) instead updates w using a randomly selected subset of the data.

One iteration of the stochastic gradient method typically consists of two steps:

- Sampling an index set I_k uniformly at random from $\{1, 2, \ldots, n\}$,
- Updating the model parameters using the stochastic gradient:

$$w^{(k+1)} = w^{(k)} - \frac{\eta}{|I_k|} \sum_{i_k \in I_k} \nabla f_{i_k}(w^{(k)}), \tag{5}$$

  where η is the learning rate.

This approach reduces the computational cost per iteration and allows for scalable optimization in large datasets.

Stochastic Proximal Gradient for LASSO

In this problem, we apply the stochastic proximal gradient to solve the LASSO problem (1).

(a) (3pt) Recall the original motivation of LASSO is to find a sparse weight w. To be precise, to find w with small l_0 norm (not really a norm),

$$\|w\|_0 = \sum_{j=1}^{d} \mathbf{1}_{w_j \neq 0}.$$

Prove that $\|\cdot\|_0$ is non-convex in $\mathbb{R}^d$. Compute $\text{prox}_{\lambda\|\cdot\|_0}$ for $d = 1$.
(Remark: Note that $\|\cdot\|_0$ is non-convex, $\text{prox}_{\lambda\|\cdot\|_0}$ may not be well-defined at some points.)

(b) (3pt) Recall that if we have a non-convex set S, we can consider the convex hull $\text{conv}(S)$, which is the smallest convex set that contains S. For non-convex functions, we can also consider the convex relaxation of it.

Assume that h is a non-convex function defined on a convex set C, a convex relaxation of h on C is a convex function r such that $r(x) \leq h(x)$ holds for any $x \in C$.

Prove that $\|\cdot\|_1$ is a convex relaxation of $\|\cdot\|_0$ on $C = \{w : \max_j |w_j| \leq 1\}$ and compute $\text{prox}_{\lambda\|\cdot\|_1}$.

(Remark: $\|\cdot\|_1$ is actually the tightest convex relaxation (convex envelope) of $\|\cdot\|_0$.)

(c) (4pt) Consider an optimization problem

$$\min_{w\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n}[f_i(w) + r(w)]. \tag{6}$$

We can combine stochastic gradient and proximal gradient and use stochastic proximal gradient to solve it. To be precise, one iteration of the stochastic proximal gradient method typically consists of two steps:

- Sampling an index set I_k uniformly at random from $\{1, 2, \ldots, n\}$,
- Updating the model parameters using the stochastic gradient:

$$w^{(k+1)} = \text{prox}_{\eta r}\left(w^{(k)} - \frac{\eta}{|I_k|}\sum_{i_k \in I_k} \nabla f_{i_k}(w^{(k)})\right), \tag{7}$$

  where η is the learning rate.

Rewrite the LASSO problem (1) in the form of (6). Specifically, identify the function f_i and r in the context of LASSO. Based on that, rewrite the update (7) in the context of LASSO.

(d) (10pt) Use the provided code to generate a training dataset `X,y`, where each row corresponds to a training data (x_i, y_i). The ground truth is `w_true`.

(d.1) Fill in the functions `soft_thresholding`, `lasso_objective` and `stochastic_proximal_gradient`.

- `soft_thresholding` takes `x, threshold` as input and $\mathrm{prox}_{\texttt{threshold}\|\cdot\|_1}(\texttt{x})$ as output.
- `lasso_objective` takes `X, y, w, lambda_reg` as input and

$$\frac{1}{2n}\|\texttt{Xw}-\texttt{y}\|_2^2+\texttt{lambda\_reg}\|\texttt{w}\|_1$$

  as output.
- `stochastic_proximal_gradient` applies the stochastic proximal gradient to LASSO.

(d.2) Use the provided hyper-parameters and codes and your functions to implement stochastic gradient descent for LASSO and obtain the numerical solution of `w_num` and the history of objective function values.

Report the following values:

- error of weight w: $\|\texttt{w\_num}-\texttt{w\_true}\|_2$
- relative error of weight w: $\dfrac{\|\texttt{w\_num}-\texttt{w\_true}\|_2}{\|\texttt{w\_true}\|_2}$.
- error of prediction y: $\|\texttt{X w\_num}-\texttt{y}\|_2$
- relative error of prediction y: $\dfrac{\|\texttt{X w\_num}-\texttt{y}\|_2}{\|\texttt{y}\|_2}$.

Plot the following results:

- `w_true` and `w_num` on the same plot.
- `objs` v.s. the number of epochs.