CAP 6610 Machine Learning, Spring 2020

Homework 2 Solution

1. (10 points) What is the distance between two parallel hyperplanes $\{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{a}^\top \boldsymbol{x} = b_1\}$ and $\{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{a}^\top \boldsymbol{x} = b_2\}$? *Hint.* Let $\boldsymbol{a}^\top \boldsymbol{x}_1 = b_1$, $\boldsymbol{a}^\top \boldsymbol{x}_2 = b_2$, and minimize $\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2$.

   Solution. The distance between two sets is the smallest distance between two points from each sets. It can be formulated as the following optimization problem

$$\begin{array}{ll} \underset{\boldsymbol{x}_1, \boldsymbol{x}_2}{\text{minimize}} & \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2 \\ \text{subject to} & \boldsymbol{a}^\top \boldsymbol{x}_1 = b_1, \boldsymbol{a}^\top \boldsymbol{x}_2 = b_2. \end{array} \tag{1}$$

   The two constraints implies that

$$\boldsymbol{a}^\top (\boldsymbol{x}_1 - \boldsymbol{x}_2) = b_1 - b_2,$$

   together with the Cauchy-Schwarz inequality

$$|\boldsymbol{a}^\top (\boldsymbol{x}_1 - \boldsymbol{x}_2)| \leq \|\boldsymbol{a}\| \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|,$$

   we see that

$$\|\boldsymbol{x}_1 - \boldsymbol{x}_2\| \leq \frac{|b_1 - b_2|}{\|\boldsymbol{a}\|},$$

   for any $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ that satisfy $\boldsymbol{a}^\top \boldsymbol{x}_1 = b_1$ and $\boldsymbol{a}^\top \boldsymbol{x}_2 = b_2$.

   Furthermore, if we let

$$\boldsymbol{x}_1 = \boldsymbol{a} \frac{b_1}{\|\boldsymbol{a}\|}, \quad \boldsymbol{x}_2 = \boldsymbol{a} \frac{b_2}{\|\boldsymbol{a}\|}, \tag{2}$$

   then

$$\|\boldsymbol{x}_1 - \boldsymbol{x}_2\| = \frac{|b_1 - b_2|}{\|\boldsymbol{a}\|}.$$

   This means (2) is a solution to Problem (1), and the distance is

$$\frac{|b_1 - b_2|}{\|\boldsymbol{a}\|}$$

2. (20 points) Let x be a real-valued random variable with sample space $\{a_1, \ldots, a_k\}$ where $a_1 \leq a_2 \leq \cdots \leq a_k$. This can be view as a categorical random variable with each catogory assigned a real value. Let $\Pr[x = a_i] = p_i$, then the vector $\boldsymbol{p}$ satisfies $\boldsymbol{p} \geq 0$ and $\boldsymbol{1}^\top \boldsymbol{p} = 1$, i.e., it lies in the probability simplex Δ. For each of the following functions of $\boldsymbol{p}$ on the probability simplex, determine if the function is convex, concave, or neither.

(a) $\mathrm{E}[x]$

(b) $\Pr[x > \alpha]$

(c) $\Pr[\alpha < x < \beta]$

(d) $-\sum_{i=1}^{k} p_i \log p_i$, the entropy of this distribution

(e) $\operatorname{var}(x)$

Solution.

(a)

$$\mathrm{E}[x] = \sum_{i=1}^{k} a_i p_i = \boldsymbol{a}^\top \boldsymbol{p},$$

where $\boldsymbol{a} = (a_1, \ldots, a_k)$. This is a linear function of $\boldsymbol{p}$, and thus both convex and concave.

(b)

$$\Pr[x > \alpha] = \sum_{i: a_i > \alpha} p_i = \boldsymbol{b}^\top \boldsymbol{p},$$

where $\boldsymbol{b}$ is defined as

$$b_i = \begin{cases} 0 & a_i \le \alpha \\ 1 & a_i > \alpha. \end{cases}$$

This is a linear function of $\boldsymbol{p}$, and thus both convex and concave.

(c)

$$\Pr[x > \alpha] = \sum_{i: \alpha < a_i < \beta} p_i = \boldsymbol{c}^\top \boldsymbol{p},$$

where $\boldsymbol{c}$ is defined as

$$c_i = \begin{cases} 1 & \alpha < a_i < \beta \\ 0 & \text{otherwise}. \end{cases}$$

This is a linear function of $\boldsymbol{p}$, and thus both convex and concave.

(d) $u \log u$ is a convex function of x, therefore $-\sum_{i=1}^{k} p_i \log p_i$ is a concave function of $\boldsymbol{p}$.

(e) We have

$$\operatorname{var}(x) = \mathrm{E}[x^2] - (\mathrm{E}[x])^2 = \sum_{i=1}^{k} a_i^2 p_i - (\boldsymbol{a}^\top \boldsymbol{p})^2.$$

This is a quadratic function with the Hessian matrix $-\boldsymbol{a}\boldsymbol{a}^\top$, which is negative semidefinite. Therefore it is a concave function of $\boldsymbol{p}$.

3. (10 points) Show that the following two convex problems are equivalent. Carefully explain how the solution of (b) is obtained from the solution of (a).

(a) The robust least squares problem

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \quad \sum_{i=1}^{n} h(\boldsymbol{\phi}_i^\top \boldsymbol{\theta} - \psi_i),$$

where $h : \mathbb{R} \to \mathbb{R}$ is the Huber function defined (with a constant M) as

$$h(t) = \begin{cases} t^2 & |t| \leq M \\ M(2|t| - M) & |t| > M. \end{cases}$$

(b) The quadratic program

$$\begin{aligned} \underset{\boldsymbol{\theta},\boldsymbol{u},\boldsymbol{v}}{\text{minimize}} \quad & \sum_{i=1}^{n} (u_i^2 + 2Mv_i) \\ \text{subject to} \quad & -\boldsymbol{u} - \boldsymbol{v} \leq \boldsymbol{\Phi}\boldsymbol{\theta} - \boldsymbol{\psi} \leq \boldsymbol{u} + \boldsymbol{v} \\ & 0 \leq \boldsymbol{u} \leq M\boldsymbol{1}, \qquad \boldsymbol{v} \geq 0. \end{aligned}$$

Solution. Suppose we fix $\boldsymbol{\theta}$ in Problem (b). First we notice that at the optimum of (b) we must have $u_i + v_i = |\boldsymbol{\phi}_i^\top \boldsymbol{\theta} - \psi_i|$, because otherwise we can further decrease the objective function without violating the constraints. Therefore $v_i = |\boldsymbol{\phi}_i^\top \boldsymbol{\theta} - \psi_i| - u_i$ at optimum.

Eliminating $\boldsymbol{v}$ yields the following problem

$$\begin{aligned} \underset{\boldsymbol{u}}{\text{minimize}} \quad & \sum_{i=1}^{n} \Big(u_i^2 - 2Mu_i + 2M|\boldsymbol{\phi}_i^\top \boldsymbol{\theta} - \psi_i| \Big) \\ \text{subject to} \quad & 0 \leq u_i \leq \min(M, |\boldsymbol{\phi}_i^\top \boldsymbol{\theta} - \psi_i|), \quad i = 1, \ldots, n. \end{aligned}$$

The problem is separable over each u_i, and we rewrite it as

$$\begin{aligned} \underset{u_i}{\text{minimize}} \quad & (u_i - M)^2 - M^2 + 2M|\boldsymbol{\phi}_i^\top \boldsymbol{\theta} - \psi_i| \\ \text{subject to} \quad & 0 \leq u_i \leq \min(M, |\boldsymbol{\phi}_i^\top \boldsymbol{\theta} - \psi_i|). \end{aligned}$$

It is easy to see that if $M < |\boldsymbol{\phi}_i^\top \boldsymbol{\theta} - \psi_i|$, then we should choose $u_i = M$ to minimize the objective; otherwise we should choose u_i to be as close to M as possible, which is $|\boldsymbol{\phi}_i^\top \boldsymbol{\theta} - \psi_i|$. Thus, we conclude that for a fixed $\boldsymbol{\theta}$ in Problem (b), the optimal value is given by the Huber function

$$\sum_{i=1}^{n} h(\boldsymbol{\phi}_i^\top \boldsymbol{\theta} - \psi_i).$$

4. (30 points) We test the performance of three regression methods on the wine data set `http://archive.ics.uci.edu/ml/datasets/Wine+Quality`. We will only consider the red wine data set, with 1599 samples. We use the first 1400 samples for training, and the last 199 samples for testing. The goal is to build a linear model of the first 11 features (together with a constant term) to predict the quality of the wine. All models are trained by solving the following optimization problem

$$\underset{\boldsymbol{w},\beta}{\text{minimize}} \quad \sum_{i=1}^{n} \ell(\boldsymbol{x}_i^\top \boldsymbol{w} + \beta - y_i),$$

where the loss functions are

- least squares loss $\ell(t) = t^2$
- Huber loss defined in the previous problem, with $M = 1$
- hinge (deadzone-linear) loss

$$\ell(t) = \begin{cases} 0 & |t| \leq 0.5 \\ |t| - 0.5 & |t| > 0.5 \end{cases}$$

The least squares loss can be directly solved by the command `Phi\y` for some properly defined `Phi`. For the latter two, you will use the `cvx` package found on Prof. Boyd's website `https://web.stanford.edu/~boyd/software.html`. Report their prediction performance on the test set using a different metric, mean absolution error (MAE), defined as $(1/n)\sum_{i=1}^{n} |y_i - \hat{y}_i|$.

Solution. The returned MAEs on the test set are 0.5330, 0.5327, and 0.5481, respectively. We see that their performances are essentially the same. Since the given scores are only integers, an average deviation of ± 0.5 is reasonably good, but not particularly impressive.

5. (30 points) We test the performance of three classification methods on the ionosphere data set `https://archive.ics.uci.edu/ml/datasets/ionosphere`. There are 351 samples. We use the first 300 samples for training, and the last 51 samples for testing. The goal is to build a linear model of the 34 features (together with a constant term) to predict the binary (± 1) outcome. All models are trained by solving the following optimization problem

$$\underset{\boldsymbol{w},\beta}{\text{minimize}} \quad \sum_{i=1}^{n} \ell(\boldsymbol{x}_i^\top \boldsymbol{w} + \beta, y_i),$$

where the loss functions are

- least squares loss $\ell(t, y) = (yt - 1)^2$
- logistic loss $\ell(t, y) = \log(1 + \exp(-yt))$
- hinge loss $\ell(t, y) = \max(0, 1 - yt)$

Again, you will use the backslash command to solve for the first model, and `cvx` to solve for the latter two. Report their prediction accuracy on the test set.

Solution. The returned prediction accuracies on the test set are all 100% correct. This is perhaps because of the fact that all of the last 51 samples are in the "good" category, which makes it somewhat easier to guess.