

**MACHINE LEARNING FOR COMPUTATIONAL FINANCE
ASSIGNMENT 1**

KARTHIK IYER

Problem 1: Linear Regression

Consider a quadratic function,

$$f(x) = \frac{1}{2} \|F x - r\|^2.$$

- (1) What is the gradient ∇f and Hessian $\nabla^2 f$ of this function? Is ∇f Lipschitz continuous? If it is, what is the Lipschitz constant?

Solution: Let $\langle \cdot, \cdot \rangle$ denote the usual inner product in the Euclidean space. Before we proceed, let us state two facts.

- (i) For any symmetric $n \times n$ matrix A , if $g(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ is defined as $g(x) = \langle Ax, x \rangle$, then $\nabla g(x) = 2Ax$. We can see why this is true by expanding the inner product and doing some algebra.
- (ii) For any $m \times n$ matrix A , if $h(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ is defined as $h(x) = \langle Ax, b \rangle$ for some $b \in \mathbb{R}^m$, then $\nabla h(x) = A^T b$. We can see why this is true by expanding the inner product and doing some algebra.

Now, since $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ is defined as $f(x) = \frac{1}{2} \|F x - r\|^2$, we can re-write $f(x)$ as $f(x) = \frac{1}{2} (\langle F x, F x \rangle - 2 \langle F x, r \rangle + \|r\|^2) = \frac{1}{2} (\langle F^T F x, x \rangle - 2 \langle F x, r \rangle + \|r\|^2)$.

We can now invoke the two facts proved above to conclude that $\nabla f(x) = F^T(F x - r)$. (Here F is a $m \times n$ matrix, $x \in \mathbb{R}^n$ and $r \in \mathbb{R}^m$).

And hence, the Hessian $\nabla^2 f = F^T F$. It is also easy to see that ∇f is Lipschitz since

$$\begin{aligned} \|\nabla f(x) - \nabla f(y)\|_2 &= \|F^T F(x - y)\|_2 \text{ for any } x, y \in \mathbb{R}^n \\ &\leq \|F^T F\|_2 \|x - y\|_2 \text{ (By definition of matrix 2-norm)} \end{aligned}$$

Thus, ∇f is Lipschitz continuous with Lipschitz constant $\|F^T F\|_2 = \|F\|_2^2$.

(Note to self: Recall that the singular values of a $m \times n$ matrix X are the square roots of the eigenvalues of the $n \times n$ matrix $X^* X$ (where $*$ stands for the transpose-conjugate matrix if it has complex coefficients, or the transpose if it has real coefficients). Thus, if X is $n \times n$ real symmetric matrix with non-negative eigenvalues, then eigenvalues and singular values coincide, but it is not generally the case). ■

- (2) If we add a quadratic penalty to the function, $f(x) = \frac{1}{2} \|F x - r\|^2 + \frac{\lambda}{2} \|x\|^2$. Answer the same set of questions in part (1) for this f .

Solution: We proceed as in part (1) to get $\nabla f(x) = F^T(F x - r) + \lambda x$ and $\nabla^2 f(x) = F^T F + \lambda I_{n \times n}$. In this case too, $\nabla f(x)$ is Lipschitz with Lipschitz constant $\|F\|_2^2 + \lambda$. ■

(3) Implement gradient descent algorithm in **Jupyter Notebook** to solve the problem,

$$\min_x \frac{1}{2} \|\mathbf{F}x - \mathbf{r}\|^2 + \frac{\lambda}{2} \|x\|^2.$$

Problem 2: LASSO

Consider LASSO objective,

$$\min_x f(x) := \frac{1}{2} \|\mathbf{F}x - \mathbf{r}\|^2 + \lambda \|x\|_1.$$

(1) Is f a β -smooth function? If it is, what is β ? If not explain the reasons.

Solution: f is not a β -smooth function for any $\beta > 0$. Had it been so, then since $\frac{1}{2} \|\mathbf{F}x - \mathbf{r}\|^2$ is β smooth for $\beta = \sigma_1^2$, $\|x\|_1$ will also be β' smooth for some β' . But $\|x\|_1$ is not C^1 . ■

(2) Consider a simpler version of the problem,

$$\min_x \frac{1}{2\eta} \|x - y\|^2 + \lambda \|x\|_1$$

what is the solution (in closed form)?

Solution: We wish to minimize $\frac{1}{2\eta} \|x - y\|_2^2 + \lambda \|x\|_1$. Note that

$$\begin{aligned} & \min_x \frac{1}{2\eta} \|y - x\|_2^2 + \lambda \|x\|_1 \\ &= \min_x \frac{1}{2\eta} [\|y\|^2 + \|x\|^2 - 2\langle x, y \rangle] + \lambda \|x\|_1 \\ &= \min_x \frac{1}{2\eta} [\|x\|^2 - 2\langle x, y \rangle] + \lambda \|x\|_1 \text{ as } y \text{ is independent of } x \\ &= \min_x \frac{1}{2\eta} [\sum_i x_i^2 - 2x_i y_i] + \lambda \sum_i |x_i| \\ &= \min_x \frac{1}{\eta} \left[\sum_i \left(\frac{x_i^2}{2} - x_i y_i + \lambda \eta |x_i| \right) \right]. \end{aligned} \tag{0.1}$$

Note that we can minimize each term inside the sum separately as the terms are essentially independent from each other. Consider the problem of minimizing $L_i(x_i) = \left(\frac{x_i^2}{2} - x_i y_i + \lambda \eta |x_i| \right)$.

Note that if $y_i > 0$, then $x_i \geq 0$ (for if $x_i < 0$; then $L_i \geq 0 = L_i(0)$. Similarly, if $y_i < 0$, then $x_i \leq 0$.

If $y_i > 0$; then since $x_i \geq 0$, $L_i = \frac{x_i^2}{2} - x_i y_i + \lambda \eta x_i$. Minimizing this quantity (by setting the first derivative to 0) gives us $x_i = y_i - \lambda \eta$. Since this is feasible only when $x_i \geq 0$ and $y_i > 0$, we get $x_i = \text{sgn}(y_i)(|y_i| - \lambda \eta)^+$.

If $y_i \leq 0$; then since $x_i \leq 0$, $L_i = \frac{x_i^2}{2} - x_i y_i - \lambda \eta x_i$. Minimizing this quantity (by setting the first derivative to 0) gives us $x_i = y_i + \lambda \eta$. Since this is feasible only when $x_i \leq 0$ and $y_i < 0$, we get $x_i = \text{sgn}(y_i)(|y_i| - \lambda \eta)^+$.

Thus in both cases we obtain $x_i = \text{sgn}(y_i)(|y_i| - \lambda\eta)^+$. Hence the desired minimizer is

$$x = (x_1, \dots, x_n) \text{ where } x_i = \text{sgn}(y_i)(|y_i| - \lambda\eta)^+.$$

■

(3) Implement proximal gradient descent algorithm in **Jupyter Notebook**.

See the accompanying Jupyter notebook for an implementation of the proximal gradient descent algorithm.

Problem 3: Robust Regression

Consider Huber objective,

$$\min_x f(x) := \rho_\kappa(\mathbf{F}x - \mathbf{r}) + \lambda\|x\|_1$$

where ρ_κ is Huber function,

$$\rho_\kappa(a) = \sum_{i=1}^m \begin{cases} \kappa|a_i| - \kappa^2/2, & |a_i| > \kappa \\ a_i^2/2, & |a_i| \leq \kappa \end{cases}$$

(1) For scalar case $a \in \mathbb{R}$, show that $\rho_\kappa(a) = \min_x \frac{1}{2}(x - a)^2 + \kappa|x|$. (Same derivation with Problem 2 (2)).

Proof. Let $p_\kappa(a) = \min_x \frac{1}{2}(x - a)^2 + \kappa|x|$. By Problem 2 part (2) we obtain $p_\kappa(a) = \frac{1}{2}([\text{sgn}(a)(|a| - \kappa)^+ - a]^2 + \kappa(|a| - \kappa)^+)$. For $|a| \geq \kappa$; we get

$$p_\kappa(a) = \frac{1}{2}\kappa^2 + \kappa|a - \text{sgn}(a)\kappa| = \frac{1}{2}\kappa^2 + \kappa|a| - \kappa = \kappa|a| - \frac{1}{2}\kappa^2.$$

For $|a| < \kappa$, we get $p_\kappa(a) = \frac{a^2}{2}$. This agrees with the definition of $\rho_\kappa(a)$. ■

(2) Is ρ_κ a β -smooth function? If it is what is β ?

Solution: Yes. It is easy to see that ρ_κ is C^1 . (As $|\cdot|$ is smooth away from 0 and since

$$\frac{\partial \rho_\kappa}{\partial a_i}(a) = \begin{cases} \kappa \text{sgn}(a_i), & |a_i| > \kappa \\ a_i, & |a_i| \leq \kappa \end{cases}$$

is continuous).

We claim that $\left| \frac{\partial \rho_\kappa}{\partial a_i}(y) - \frac{\partial \rho_\kappa}{\partial a_i}(x) \right| \leq |y_i - x_i|$.

To prove this, we assume without loss of generality that $y_i > x_i$ and consider the following 6 mutually exhaustive cases:

Case 1: $y_i \geq \kappa > x_i > -\kappa$. In this case, $\left| \frac{\partial \rho_\kappa}{\partial a_i}(y) - \frac{\partial \rho_\kappa}{\partial a_i}(x) \right| = |\kappa - \kappa| = 0 \leq y_i - x_i = |y_i - x_i|$.

Case 2: $y_i \geq \kappa > -\kappa \geq x_i$. In this case, $\left| \frac{\partial \rho_\kappa}{\partial a_i}(y) - \frac{\partial \rho_\kappa}{\partial a_i}(x) \right| = |\kappa - \kappa| = 0 \leq y_i - x_i = |y_i - x_i|$.

Case 3: $y_i > x_i \geq \kappa$. In this case, $\left| \frac{\partial \rho_\kappa}{\partial a_i}(y) - \frac{\partial \rho_\kappa}{\partial a_i}(x) \right| = 0 \leq y_i - x_i = |y_i - x_i|$.

Case 4: $\kappa \geq y_i > x_i \geq -\kappa$. In this case, $\left| \frac{\partial \rho_\kappa}{\partial a_i}(y) - \frac{\partial \rho_\kappa}{\partial a_i}(x) \right| = |y_i - x_i| \leq |y_i - x_i|$.

Case 5: $\kappa \geq y_i > -\kappa \geq x_i$. In this case, $\left| \frac{\partial \rho_\kappa}{\partial a_i}(y) - \frac{\partial \rho_\kappa}{\partial a_i}(x) \right| = y_i + \kappa \leq y_i - x_i = |y_i - x_i|$.

Case 6: $-\kappa \geq y_i > x_i$. In this case, $\left| \frac{\partial \rho_\kappa}{\partial a_i}(y) - \frac{\partial \rho_\kappa}{\partial a_i}(x) \right| = 0 \leq |y_i - x_i|$.

Our claim is hence justified. This claim in particular proves that ρ_κ is β smooth with $\beta = 1$. ■

(3) Is $\rho_\kappa(\mathbf{F}x - \mathbf{r})$ a β -smooth function with respect to x ? If it is what is β ?

Proof. Yes. Firstly, the composition of two C^1 functions is C^1 . Moreover, by chain rule and the fact that $\rho_\kappa(a)$ is 1 smooth, we see that by chain rule, that for $h(x) = \rho_\kappa(\mathbf{F}x - \mathbf{r})$, $\nabla h(x) = \mathbf{F}^T \nabla \rho_\kappa(\mathbf{F}x - \mathbf{r})$. Hence

$$\|\nabla h(x) - \nabla h(y)\|_2 \leq \|\mathbf{F}^T\|_2 \|\mathbf{F}\|_2 \|x - y\|_2.$$

Thus $\rho_\kappa(\mathbf{F}x - \mathbf{r})$ is $\|\mathbf{F}\|_2^2$ smooth. ■

(4) Implement proximal gradient descent method in **Jupyter Notebook**.

Problem 4: Logistic Regression

Assume we only care about distinguishing assets that will go up or go down, consider logistic regression objective,

$$\min_x f(x) := \sum_{i=1}^m \{ \log(1 + \exp(\langle f^i, x \rangle)) - s^i \langle f^i, x \rangle \} + \frac{\lambda}{2} \|x\|^2$$

where $s^i \in \{-1, 1\}$, is the indicator of if the asset will go up or go down. Let's also denote $\mathbf{F} = [f^1, \dots, f^m]^T$ as we do in the note.

(1) Calculate the gradient of f , namely ∇f . Is f a β -smooth function? If it is, what is the β ?

Solution: We note that $\frac{\partial f}{\partial x_j} = \sum_{i=1}^m \left(\frac{f_j^i \exp(\langle f^i, x \rangle)}{1 + \exp(\langle f^i, x \rangle)} - s^i f_j^i \right) + \lambda x_j$ for $j = 1, 2, \dots, n$.

Let $r = \mathbf{F}x$. Note that $\mathbf{F} \in \mathcal{M}_{m \times n}$ and $x \in \mathbb{R}^n$. Let $r = (r_1, r_2, \dots, r_m)$. Define $\exp(r) = (\exp(r_1), \exp(r_2), \dots, \exp(r_m))$.

$$\nabla f(x) = \mathbf{F}^T \left(\frac{1}{1 + \exp(-r)} - s \right) + \lambda x, \quad (0.2)$$

where $\frac{1}{1 + \exp(r)}$ is gotten by componentwise operations.

After some algebraic manipulations, we can bound $\|\nabla^2 f\| \leq \|\mathbf{F}\|_2^2 + \lambda$. Thus, f is β smooth with $\beta = \|\mathbf{F}\|_2^2 + \lambda$. (Note that this is the same β as we had for OLS regression. I am not sure if this is the optimal β though.) ■

(2) Implement gradient descent method in **Jupyter Notebook** over training data. Do cross validation to the best λ , and report the test error.

Solution: See the accompanying Jupyter notebook. The test error turns out to be 47.2%. ■