Problem 1

(a) Supervised Learning learns from labeled examples to make predictions, while Unsupervised Learning discovers patterns in data without any guidance from labels.

(b) 2) True

(c)

(c) data matrix $X \in \mathbb{R}^{n \times d}$ has full column rank.

→ Explain why $X^T X$ is PD (Hint: linear independence).

↳ for any matrix $A \neq 0$,

$A^T \underline{X^T X} A > 0$.

$XA = 0$, only when $A = 0$.

Proof: $(X^T X)^T = X^T X$, thus X is symmetric.

Let matrix $A \in \mathbb{R}^d$, $A \neq 0$.

Because $X \in \mathbb{R}^{n \times d}$ has full column rank.

so X is linear independent,

so $XA \neq 0$, $\|XA\|_2^2 > 0$

$\|XA\|_2^2 = (XA)^T (XA) = A^T X^T X A > 0$

so $X^T X$ is Positive Definite.

Problem 2

(2.a)

Problem 2.

(a) $\min_{\theta\in\mathbb{R}^d} \|X\theta - y\|_2^2$

$= \min_{\theta\in\mathbb{R}^d} \|V\Sigma_1 U_1^T\theta - y\|_2^2$

Let $A := V\Sigma_1$ which is square matrix of full rank.

Let $z = U_1^T\theta$.

Solve: $\mathcal{L}(z) = \min_z \|Az - y\|_2^2$

$= (Az-y)^T(Az-y) = z^TA^TAz - 2z^TA^Ty + y^Ty$

$\nabla_z\mathcal{L}(z) = 2A^T(Az-y)$

set $\nabla_z\mathcal{L}(z) = 0$ $\Rightarrow$ the optimal solution $\hat{z}$ satisfies: $A^TA\hat{z} = A^Ty$

Because $A \in \mathbb{R}^{n\times n}$ is a square matrix of full rank.

$\Rightarrow A^TA \in \mathbb{R}^{n\times n}$ is full rank and invertible

We have $\hat{z} = (A^TA)^{-1}A^Ty$

so, $\hat{z} = A^{+}y \Rightarrow U_1^T\hat{\theta} = (V\Sigma_1)^{+}y = (\Sigma_1^TV^TV\Sigma_1)^{-1}\Sigma_1^TV^Ty$

$= (\Sigma_1^T\Sigma_1)^{-1}\Sigma_1^T(\Sigma_1\Sigma_1^{-1})V^Ty$

$= \Sigma_1^{-1}V^Ty$

we want to solve: $\boxed{U_1^T\hat{\theta} = \Sigma_1^{-1}V^Ty}$

this is a system of $\underline{n}$ equations in $\underline{d}$ unknowns $(d>n)$.

So, there is infinite many solutions.

$\Rightarrow$ one particular solution θ_p:

Try: $\theta_p = U_1\hat{z} = U_1\Sigma_1^{-1}V^Ty$

$U_1^T\theta_p = (U_1^TU_1)\Sigma_1^{-1}V^Ty = \Sigma_1^{-1}V^Ty$ ✓

$\Rightarrow$ homogeneous solution θ_h:

Solve: $U_1^T\theta_h = 0$

we know $U_1^TU_2 = 0$

So: $\theta_h = U_2\alpha$ (for any $\alpha \in \mathbb{R}^{d-n}$).

$\Rightarrow$ general solution:

$\hat{\theta} = \theta_p + \theta_h = U_1\Sigma_1^{-1}V^Ty + U_2\alpha \quad (\forall \alpha \in \mathbb{R}^{d-n})$

- The ~~best~~ optimal function value for $\min_{\theta\in\mathbb{R}^d}\|X\theta - y\|_2^2$

Let $\theta = \theta_p$,

Optimal function value $= \|V\Sigma_1U_1^T\theta_p - y\|_2^2$

$= \|V\Sigma_1U_1^TU_1\Sigma_1^{-1}V^Ty - y\|_2^2$

$= \|y-y\|_2^2 = 0$

.

(2.b)

(b). Solve $\min_{\theta \in \mathbb{R}^d} \|X\theta - y\|_2^2 + \lambda \|\theta\|_2^2$

Let $J(\theta) = \|X\theta - y\|_2^2 + \lambda \|\theta\|_2^2$

$$= (X\theta - y)^T (X\theta - y) + \lambda \theta^T \theta$$

$$= \theta^T X^T X \theta - 2\theta^T X^T y + y^T y + \lambda \theta^T \theta$$

$$= \theta^T (X^T X + \lambda I)\theta - 2\theta^T X^T y + y^T y$$

$\nabla_\theta J(\theta) = 2(X^T X + \lambda I)\theta - 2X^T y$

Set $\nabla_\theta J(\theta) = 0 \Rightarrow \boxed{(X^T X + \lambda I)\theta = X^T y}$

Because $X \in \mathbb{R}^{n \times d}$, $n < d$

$\Rightarrow X^T X$ is ~~se~~ positive semi-definite.

note that $\lambda > 0$.

therefore $(\lambda I + X^T X)$ is positive definite.
and invertible.

so. $\hat{\theta} = (X^T X + \lambda I)^{-1} X^T y$

Problem 3

(3.a)

Problem 3.

(a) $y = X\theta^* + \epsilon \Rightarrow \epsilon = y - X\theta^*$

$\Rightarrow \varepsilon_i = y_i - x_i^T\theta$

$\varepsilon_i \sim \mathcal{L}(0, b)$, $p(\varepsilon_i) = \frac{1}{2b} e^{-\frac{|\varepsilon_i|}{b}}$, $b > 0$.

So, $p(y_i | x_i, \theta) = \frac{1}{2b} \exp\left(-\frac{|y_i - x_i^T\theta|}{b}\right)$

since the data are independent:

$$p(y|X,\theta) = \prod_{i=1}^{n} p(y_i|x_i,\theta) = \prod_{i=1}^{n} \frac{1}{2b} \exp\left(-\frac{|y_i - x_i^T\theta|}{b}\right)$$

$$\mathcal{L}(\theta) = \log p(y|X,\theta) = \sum_{i=1}^{n} \log\left(\frac{1}{2b} \exp\left(-\frac{|y_i - x_i^T\theta|}{b}\right)\right)$$

$$= \sum_{i=1}^{n} \left(-\log(2b) - \frac{|y_i - x_i^T\theta|}{b}\right)$$

$$= -n\log(2b) - \frac{1}{b}\sum_{i=1}^{n} |y_i - x_i^T\theta|$$

$$\hat{\theta} = \arg\max_\theta \mathcal{L}(\theta)$$

$$= \arg\max_\theta \left(-\frac{1}{b}\sum_{i=1}^{n} |y_i - x_i^T\theta|\right)$$

$$= \arg\min_\theta \sum_{i=1}^{n} |y_i - x_i^T\theta| = \arg\min_\theta \|y - X\theta\|_1$$

(3.b)

(b) $\nabla_\theta \mathcal{L}(\theta) = \nabla_\theta H_\mu(X\theta - y)$

Let $r = X\theta - y \Rightarrow \nabla_\theta \mathcal{L}(\theta) = X^T \nabla_r H_\mu(r) \Big|_{r = X\theta - y}$

$$= X^T \nabla_r \left(\sum_{j=1}^{n} h_\mu(r_j)\right)$$

$$\nabla_r H_\mu(r) = \begin{bmatrix} \frac{d}{dr_1} h_\mu(r_1) \\ \frac{d}{dr_2} h_\mu(r_2) \\ \vdots \\ \frac{d}{dr_j} h_\mu(r_j) \end{bmatrix}$$

$$h_\mu(z) = \begin{cases} |z| & , |z| \geq \mu \\ \frac{z^2}{2\mu} + \frac{\mu}{2} & , |z| \leq \mu . \end{cases}$$

$$\Rightarrow \frac{d}{dz} h_\mu(z) = \begin{cases} \text{sign}(z), & |z| \geq \mu . \\ \frac{z}{\mu} , & |z| \leq \mu \end{cases}$$

Let $g_j = \frac{d}{dr_j} h_\mu(r_j) = \begin{cases} \text{sign}(r_j) , & |r_j| \leq \mu \\ \frac{r_j}{\mu} , & |r_j| \geq \mu . \end{cases}$

$\therefore \nabla_r H_\mu(r) = g .$

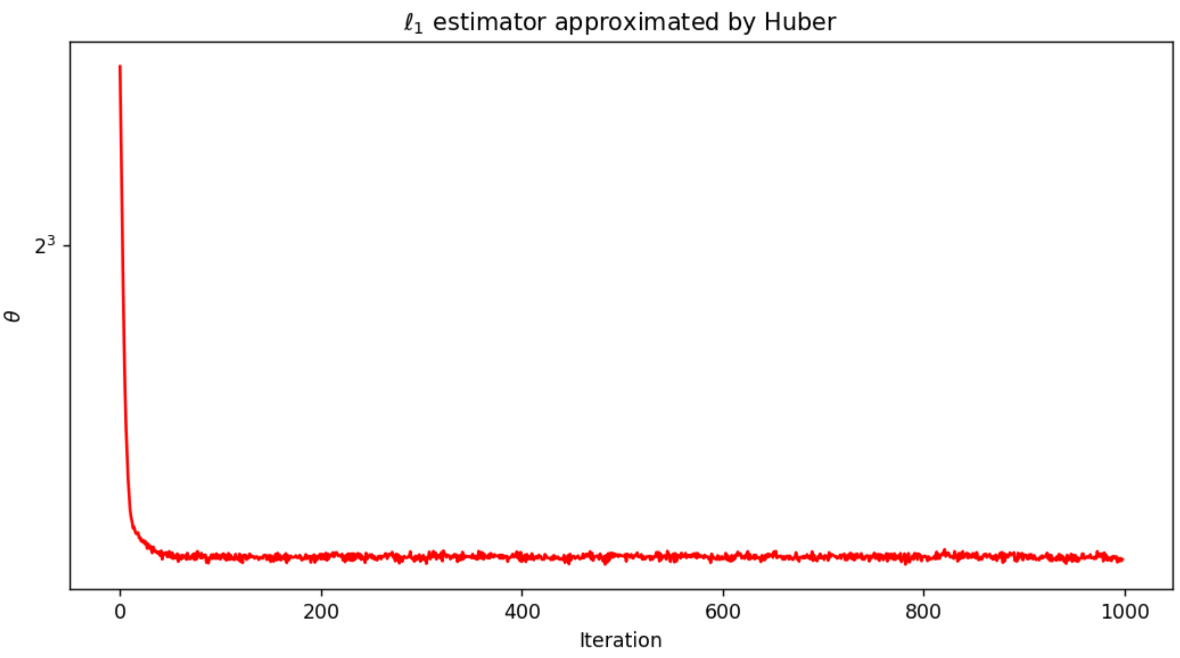
Now, $\underline{\nabla_\theta \mathcal{L}(\theta) = X^T \nabla_r H_\mu(r) = X^T g}$

where $g_j = \begin{cases} \text{sign}(r_j) , & |r_j| \leq \mu \\ \frac{r_j}{\mu} , & |r_j| \geq \mu . \end{cases}$

$r = X\theta - y$

(3.c)

Code: p3.py



Problem 4

Problem 4.

(a). proof: since the data is linearly separable, we have: $y_i(\theta^{*T}x_i) > 0$ for all $i=1,2,\dots n$.

Therefore, $\rho = \min_{1\le i\le n} y_i(\theta^{*T}x_i) > 0$.

Q.E.D.

(b). proof: $\theta_k = \theta_{k-1} + y_{k-1}x_{k-1}$

$$\Rightarrow \theta_k^T\theta^* = (\theta_{k-1}+y_{k-1}x_{k-1})^T\theta^*$$

$$= \theta_{k-1}^T\theta^* + y_{k-1}x_{k-1}^T\theta^* = \theta_{k-1}^T\theta^* + y_{k-1}\theta^{*T}x_{k-1}$$

Because, $y_{k-1}\theta^{*T}x_{k-1} \ge \min_{1\le k\le n} y_k(\theta^{*T}x_k) = \rho$

Therefore, $\cancel{\theta_k^T\theta^* \ge \theta_{k-1}^T\theta^*}$

$$\theta_k^T\theta^* = \theta_{k-1}^T\theta^* + y_{k-1}\theta^{*T}x_{k-1} \ge \theta_{k-1}^T\theta^* + \rho$$

Q.E.D.

We can know: $\theta_1^T\theta^* \ge \theta_0^T\theta^* + \rho = 0 + \rho = \rho$

$\theta_2^T\theta^* \ge \theta_1^T\theta^* + \rho \ge 2\rho$

$\vdots$

By induction:

$$\theta_k^T\theta^* \ge k\rho.$$

Q.E.D.

(c) $\|\theta_k\|^2 = \|\theta_{k-1}+y_{k-1}x_{k-1}\|^2 = (\theta_{k-1}+y_{k-1}x_{k-1})^T(\theta_{k-1}+y_{k-1}x_{k-1})$

$$= (\theta_{k-1}^T + y_{k-1}x_{k-1}^T)(\theta_{k-1} + y_{k-1}x_{k-1})$$

$$= \theta_{k-1}^T\theta_{k-1} + 2y_{k-1}(\theta_{k-1}^T x_{k-1}) + y_{k-1}^2 x_{k-1}^T x_{k-1}$$

$$= \|\theta_{k-1}\|^2 + 2y_{k-1}(\theta_{k-1}^T x_{k-1}) + y_{k-1}^2\|x_{k-1}\|^2$$

Note: $y_{k-1}^2 = 1$, since $y_i \in \{-1,1\}$.

$y_{k-1}\theta_{k-1}^T x_{k-1} < 0$, because x_{k-1} is misclassified.

Therefore $\|\theta_k\|^2 \le \|\theta_{k-1}\|^2 + \|x_{k-1}\|^2$

Q.E.D.

(d). from question (4.c):

$$\|\theta_k\|^2 \le \|\theta_{k-1}\|^2 + \|x_{k-1}\|^2 \le \|\theta_{k-1}\|^2 + R^2$$

$$(R = \max_{1\le i\le n} \|x_i\|)$$

Apply recursively: $\|\theta_1\|^2 \le \|\theta_0\|^2 + R^2 = 0 + R^2$

$$\|\theta_2\|^2 \le \|\theta_1^2\| + R^2 \le 2R^2$$

$$\vdots$$

$$\|\theta_k\|^2 \le kR^2$$ Q.E.D.

(e). from question (4.b): $\theta_k^T\theta^* \ge k\rho$

from (4.d): $\|\theta_k\|^2 \le kR^2$

$$\Rightarrow \frac{\theta_k^T\theta^*}{\|\theta_k\|} \ge \frac{k\rho}{\sqrt{kR^2}} = \frac{k\rho}{\sqrt{k}R} = \sqrt{k}\cdot\frac{\rho}{R}$$ Q.E.D.

recall the Cauchy-Schwarz inequality:

$$\theta_k^T\theta^* \le \|\theta_k\|\cdot\|\theta^*\| \Rightarrow \frac{\theta_k^T\theta^*}{\|\theta_k\|\cdot\|\theta^*\|} \le 1$$

$$\Rightarrow \frac{\theta_k^T\theta^*}{\|\theta_k\|} \le \|\theta^*\|$$

So, $\sqrt{k}\frac{\rho}{R} \le \frac{\theta_k^T\theta^*}{\|\theta_k\|} \le \|\theta^*\|$

Therefore, $\sqrt{k} \le \|\theta^*\|\frac{R}{\rho}$

$$\Rightarrow k \le \frac{R^2}{\rho^2}\|\theta^*\|^2$$

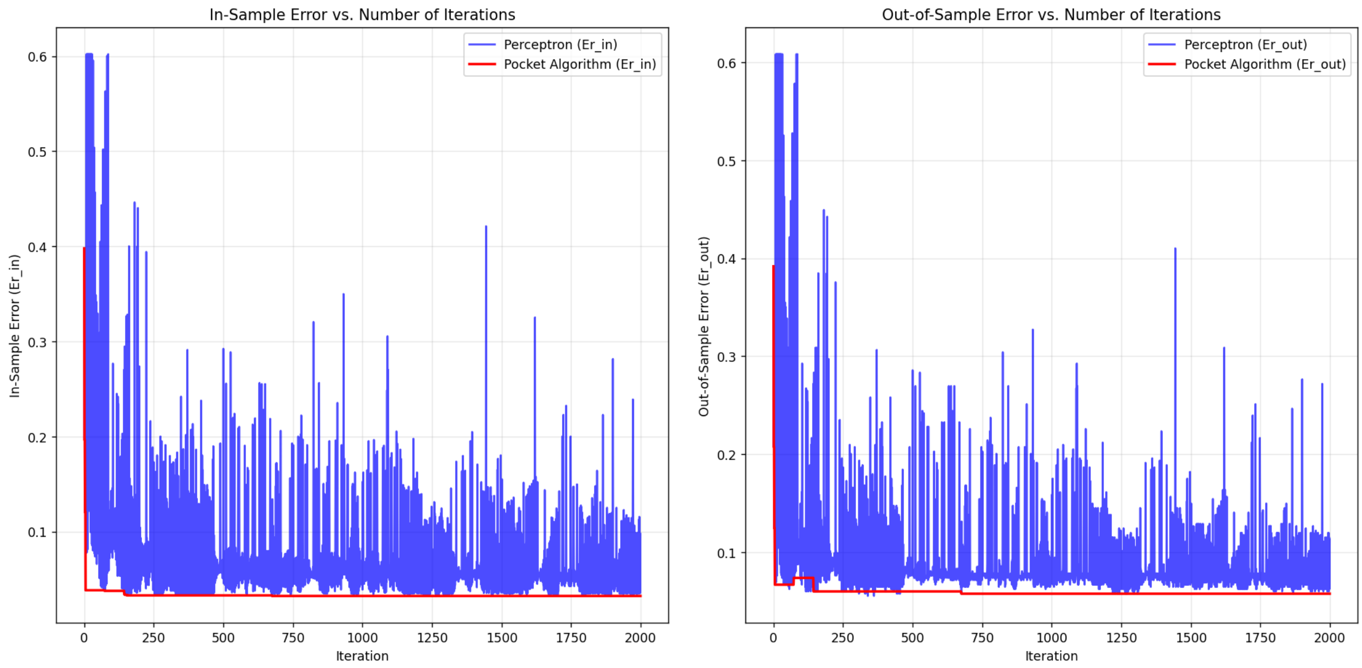
So, $\bar{k} \le \frac{R^2\|\theta^*\|^2}{\rho^2}$ Q.E.D.

Campus

Problem 5

(5.1) code: p5.py

(5.2)



Discussion of the results:

The Perceptron's errors go up and down a lot during training. This is because the data (digits "1" and "6") cannot be perfectly separated by a straight line. The Pocket Algorithm is more stable. It remembers the best solution it has found so far. Even if the Perceptron's current weights get worse, the Pocket keeps the best ones. So, its training error only gets better or stays the same.

(5.3)