Homework 1

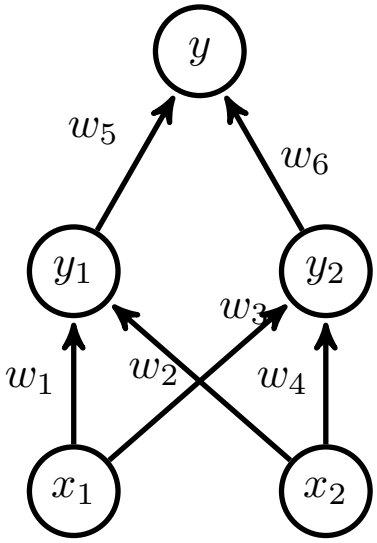
Fall 23, CS 442: Trustworthy Machine Learning
Due Friday Sep. 22nd at 23:59 CT

Instructor: Han Zhao

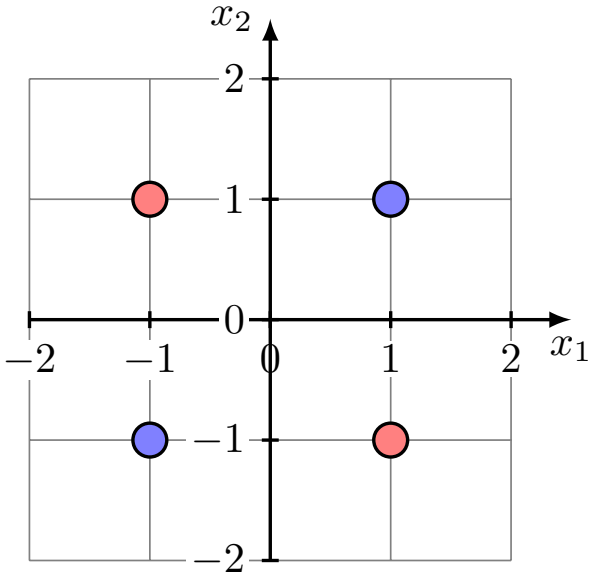
Instructions for submission All the homework submissions should be typeset in LaTeX. For all the questions, please clearly justify each step in your derivations or proofs.

1 Feed-forward Neural Networks [30 pts]

Consider a feed-forward neural network (FNN) with one input layer, one hidden layer and one output layer shown in Fig. 1a. In this problem we will use the FNN to classify the XOR pattern shown in Fig. 1b. Suppose that the activation function at every unit in the FNN are linear, i.e., $y_1 = w_1 x_1 + w_2 x_2$, $y_2 = w_3 x_1 + w_4 x_2$ and $y = w_5 y_1 + w_6 y_2 + w_0$. Classify the input instance (x_1, x_2) as $+1$ if $y(x_1, x_2) \geq 0$ otherwise -1.



(a) A three-layer feed-forward neural network with two inputs.



(b) XOR pattern. Instances in blue have label $+1$ while instances in red have label -1.

1.1 [10pts]

For any weight configuration $(w_0, w_1, w_2, w_3, w_4, w_5, w_6)$ of the FNN shown above, can you find another FNN with only two layers, i.e., one input layer with two inputs x_1, x_2 and one output layer with one output unit y that computes the same function? If yes, describe a such two-layer FNN and give the weights as functions of $(w_0, w_1, w_2, w_3, w_4, w_5, w_6)$. If no, briefly describe your reasoning.

1.2 [10pts]

Show that there is no weight configuration $(w_0, w_1, w_2, w_3, w_4, w_5, w_6)$ under which the FNN shown in Fig. 1a can classify the XOR pattern with no error.

1.3 [10pts]

For the FNN shown in Fig. 1a, will changing the activation functions at y_1 and y_2 to be nonlinear help classify the XOR pattern? If yes, construct a nonlinear activation function $f(\cdot)$ to be applied only at y_1 and y_2, i.e., $y_1(x_1, x_2) = f(w_1x_1 + w_2x_2)$, $y_2(x_1, x_2) = f(w_3x_1 + w_4x_2)$, such that the new FNN can perfectly classify the XOR pattern. If no, briefly describe your reasoning on why it is not possible.

2 The Price of Statistical Parity [30pts]

Consider a binary classification problem with two groups. Let (X, A, Y) be the tuple drawn from an underlying distribution μ. For simplicity, in this problem we assume all the variables are binary, i.e., $X, A, Y \in \{0, 1\}$. Recall from the lecture, in this example we use A to denote the group membership of an instance, i.e., $A = 0$ means the majority group whereas $A = 1$ means the minority group. More concretely, let $p \geq 1/2$ be the marginal probability of $A = 0$: $\Pr_\mu(A = 0) = p$. To complete the specification of the joint distribution μ over (X, A, Y), the group-wise distributions over the pair (X, Y) are given as follows:

- For each group $A = a \in \{0, 1\}$, the conditional probability $\Pr_\mu(X = 1 \mid A = a) = 1/2$ is uniform, i.e., with equal probabilities, X can take either 0 or 1.
- For each group $A = a \in \{0, 1\}$, $\Pr_\mu(Y = a \mid A = a) = 1$, i.e., with probability 1 the value of the target Y equals a.

2.1 [10pts]

Show that there exists a deterministic classifier $h : \{0, 1\} \times \{0, 1\} \to \{0, 1\}$, taking the pair (x, a) as input and predicts the corresponding label, is simultaneously perfect on both groups. Construct such a deterministic classifier. Note: we say a classifier to be perfect if it achieves 0 classification error on the corresponding distribution.

2.2 [10pts]

Now we consider a randomized classifier h as follows. Upon receiving the input pair (x, a), a randomized classifier $h(x, a)$ will flip a fair coin. Depending on the outcome of the coin, the randomized

classifier $h(x, a)$ will make the following prediction:

$$h(x,a) = \begin{cases} 0 & \text{If the outcome of the fair coin is H} \\ 1 & \text{If the outcome of the fair coin is T.} \end{cases}$$

2.2.1 [5pts]

What is the classification error rate of this randomized classifier on the joint distribution μ? i.e., what is $\Pr_{(X,A,Y)\sim\mu}(h(X,A) \neq Y)$?

2.2.2 [5pts]

Show that despite taking the group membership A explicitly as its input, the above randomized classifier satisfies statistical parity.

2.3 [10pts]

2.3.1 [8pts]

Prove that, for any deterministic classifier $h(X, A)$[1], if $h(X, A)$ satisfies statistical parity, then

$$\Pr_{\mu}(h(X,A) \neq Y \mid A = 0) + \Pr_{\mu}(h(X,A) \neq Y \mid A = 1) = 1.$$

2.3.2 [2pts]

Using the result above, show that the overall error rate of h over μ is at least $1 - p$, i.e., $\Pr_{\mu}(h(X, A) \neq Y) \geq 1 - p$.

3 Matrix Calculus [10pts]

A classic problem in unsupervised machine learning is known as *symmetric matrix factorization*. Given a matrix $P \in \mathbb{R}^{n\times n}$, the goal is to find a matrix $X \in \mathbb{R}^{n\times k}$ such that $P \approx XX^\top$. In this problem, we will derive the gradient of the following objective function with respect to X:

$$\min_{X\in\mathbb{R}^{n\times k}} \quad \mathcal{L}(X) := \frac{1}{2}\|P - XX^\top\|_F^2,$$

where $\|\cdot\|_F$ denotes the Frobenius norm of a matrix, i.e,. $\|A\|_F^2 = \sum_{i=1}^m \sum_{j=1}^n A_{ij}^2$ for $A \in \mathbb{R}^{m\times n}$. One typical application of the above problem is to find an embedding of n objects into a k-dimensional space, where P_{ij} denotes the similarity between the i-th and j-th objects. The above objective function encourages the similarity between the i-th and j-th objects to be approximated by the inner product between the i-th and j-th rows of X.
Please derive the gradient of $\mathcal{L}(X)$ with respect to X. Note: you need to show your derivation step by step, and you need to express the gradient in matrix notation in terms of X and P only.

[1]This actually also applies to randomized classifiers as well, but in this problem you only need to show this for deterministic classifiers.

4 The Implicit Bias of Gradient Descent in Linear Regression [30 pts]

Consider a dataset $\{(x_i, y_i)\}_{i=1}^n$ where $x_i \in \mathbb{R}^d$ is the i-th input and $y_i \in \mathbb{R}$ is the i-th label. The loss function for linear regression on this dataset is given by

$$\min_{w \in \mathbb{R}^d} \quad \mathcal{L}(w) := \frac{1}{2} \sum_{i=1}^{n} (w \cdot x_i - y_i)^2, \tag{1}$$

where $w \in \mathbb{R}^d$ is the model parameter of interest. In the course we find the optimal solution w^* via solving the *normal equation*. In this problem, instead, we will use the gradient descent method to numerically find the optimal solution instead. In particular, in order to solve (1), we apply the following algorithm:

Procedure 1 Gradient Descent for Linear Regression

Input: Initial model parameter w_0
1: **for** $t = 1, 2, \ldots$ until convergence **do**
2: $\quad w_t \leftarrow w_{t-1} - \frac{1}{t} \nabla_w \mathcal{L}(w_{t-1})$
3: **end for**
4: **return** w^*

In this problem, let's assume that the above algorithm will converge, and we use w^* to denote the convergent point.

4.1 [10pts]

Prove that $w^* - w_0 \in \text{span}\{x_1, \ldots, x_n\}$.

4.2 [10pts]

Let $X \in \mathbb{R}^{n \times d}$ be the data matrix where the i-th row of this matrix is given by $x_i^\top$, and let $y = (y_1, \ldots, y_n)^\top \in \mathbb{R}^n$ be the label vector.

4.2.1 [5pts]

Show that w^* satisfies the normal equation, i.e., $X^\top X w^* = X^\top y$.

4.2.2 [5pts]

For any vector $\widehat{w} \in \mathbb{R}^d$ that is the optimal solution of (1), show that $w^* - \widehat{w} \in \text{Ker}(X)$, where $\text{Ker}(\cdot)$ denotes the kernel of a matrix.

4.3 [10pts]

Prove that among all the model parameters that optimize (1), w^* has the minimum distance to w_0. Formally, let $W := \{\widehat{w} \in \mathbb{R}^d : \widehat{w} \text{ is an optimal solution to (1)}\}$. Prove that $w^* = \arg\min_{w \in W} \|w - w_0\|_2$. Note: this means that gradient descent finds the optimal solution that is closest to the initial point.

Dongheng Lin
September 15th, 2023

CS 442: Trustworthy Machine Learning
Homework 1

Solutions.

Q1

1.1: Given the FNN, the final outcome y can be expressed by following linear function.

$$\begin{aligned} y_1 &= w_1x_1 + w_2x_2 \\ y_2 &= w_3x_1 + w_4x_2 \\ y &= w_5y_1 + w_6y_2 \end{aligned}$$

Hence,

$$\begin{aligned} y &= y_1 + y_2 \\ &= (w_5w_1 + w_6w_3)x_1 + (w_5x_2 + w_6x_4)x_2 \end{aligned}$$

We can further represent it using a two-layer FNN with parameters above as illustrated in Figure 1

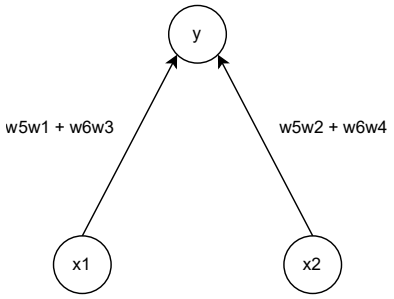


Figure 1: Equivalent 2-layer-FNN

1.2: According to **1.1**, we may notice that the decision boundary is actually a straight line in the input space. Denote an equivalent linear function of the FNN as line L $ax_1 + bx2 + c = 0$.
If two points locate on the same side of a line, it should satisfy following inequality:

$$(ax_1^1 + bx2^1 + c) * (ax_1^2 + bx2^2 + c) \quad > \quad 0$$

While for the point pairs that locates on different sides of a line, there is another inequality:

$$(ax_1^1 + bx2^1 + c) * (ax_1^2 + bx2^2 + c) \quad < \quad 0$$

According to the figure in the description, we have 2 pairs of data points that should lines on the same sides of a potential linear classifier (1, 1), (-1, -1) and (1, -1), (-1, 1). Plug that in gives:

$$\begin{aligned} (a+b+c) * (-a-b+c) &> 0, \text{ (1) Since (1, 1) and (-1, -1) lines on the same side} \\ (a+b+c) * (a+b+c) &> 0, \text{ (2) Since (1, 1) must lines on the same side of itself} \\ (a+b+c) * (a-b+c) &< 0, \text{ (3) Since (1, 1) and (1, -1) lines on the different sides} \\ (a+b+c) * (-a+b+c) &< 0, \text{ (4) Since (1, 1) and (-1, 1) lines on the different sides} \end{aligned}$$

Combining the first two and last two inequalities above gives the following contradiction:

$$\begin{aligned} 2c * (a+b+c) &> 0, \text{ adding (1) and (2)} \\ 2c * (a+b+c) &< 0, \text{ adding (3) and (4)} \end{aligned}$$

Hence, we have found a contradiction, which proves that there can not be a set of a, b, c that makes up a linear equation that classifies the XOR data points in this problem, which means, there is no $\{w_1, w_2, w_3, w_4, w_5, w_6\}$ can make the FNN correctly classifies the XOR problem.

1.3: If we use a special nonlinear sign function $\sigma(y_i)$ as the activation function, then we have:

$$\sigma(y_i) = \begin{cases} 1, & \text{if } y_i \geq -1 \\ -2, & \text{if } y_i < -1 \end{cases} \tag{1}$$

Now, let's pick proper weights $w_1, w_2, w_3, w_4, w_5, w_6$ to construct an FNN such that it correctly classifies the XOR data points.

For this problem, setting $w_1 = 1, w_2 = -1, w_3 = -1, w_4 = 1, w_5 = 1.1, w_6 = 1$ should solve the classification.

Plugging in the values:

- for input $(1, 1)$, $y(1, 1) = \sigma(1 - 1) + \sigma(-1 + 1) = 2 > 0$, output label $+1$
- for input $(-1, -1)$, $y(-1, -1) = \sigma(-1 + 1) - \sigma(1 - 1) = 2 > 0$, output label $+1$
- for input $(1, -1)$, $y(1, -1) = \sigma(1 + 1) + \sigma(-1 - 1) = -1 < 0$, output label -1
- for input $(-1, 1)$, $y(-1, 1) = \sigma(-1 - 1) + \sigma(1 + 1) = -1 < 0$, output label -1

Here, as it is shown, we have constructed a FNN that correctly classifies all the data points given.

Q2

2.1 Given that $Pr_\mu(Y = a|A = a) = 1$, the $h(x, a) = a$ always equals actual y. A fair deterministic classifier h can be a classifier with Y simply copying the input value of a as output, which is:

$$h(x, a) \quad = \quad a$$

2.2.1

It is expected to be $\frac{1}{2}$. Because that the randomized classifier $h(X, A)$does not depend on the x and a, This means for each time of testing, it has a 50% chance of getting the correct label. Consequently, the classification error rate is also 50% which can be written as:

$$Pr(X, A, Y) \sim \mu(h(X, A) \neq Y) = 0.5$$

2.2.2

The statistical parity requires that the predictions are independent of the protected attribute, which in this case is A. Under the randomized classifier, we can confirm that the probability of prediction on certain classes is the same for both values of A. i.e.

$$Pr(h(X, A = 0) = 1) = Pr(h(X, A = 1) = 1) = 0.5$$
$$Pr(h(X, A = 0) = 0) = Pr(h(X, A = 1) = 0) = 0.5$$

It clearly shows that the classifier is not taking value A in decision making and thus maintains statistical parity.

2.3.1

Considering the dataset follows a rule where $Pr_\mu(Y = a|A = a) = 1$. We know that:

$$Pr_\mu(Y = 0|A = 0) = 1$$
$$Pr_\mu(Y = 1|A = 1) = 1$$

Let's denote the error rates of the classifier on each group as:

$$e_0 = Pr_\mu(h(X, A) \neq Y|A = 0)$$
$$e_1 = Pr_\mu(h(X, A) \neq Y|A = 1)$$

Given the way the dataset is constructed, where Y equals to A all the time, we can write the above probabilities into:

$$e_0 = Pr(h(X, A = 0) = 1)$$
$$e_1 = Pr(h(X, A = 1) = 0)$$

If there is statistical parity in the classifier, given the equations above, we have:

$$Pr_\mu(h(X, A = 0) = 1) = Pr(h(X, A = 1) = 1) = e_0$$

Also, within the group A = 1, the probability of 0 is the complement of predicting 1. Hence,

$$e_1 = 1 - e_0 = 1 - Pr(h(X, A = 1) = 1) = Pr_\mu(h(X, A) \neq Y | A = 1)$$

Combining e_0 and e_1 gives:

$$e_0 + e_1 = Pr(h(X, A = 1) = 1) + 1 - Pr(h(X, A = 1) = 1) = 1$$

Thus, we can conclude that for any deterministic classifier $h^1(X, A)$ which satisfies statistical parity.

$$Pr_\mu(h(X, A) \neq Y | A = 0) + Pr_\mu(h(X, A) \neq Y | A = 1) = 1$$

2.3.3

Use the above result that:

$$Pr_\mu(h(X, A) \neq Y | A = 0) + Pr_\mu(h(X, A) \neq Y | A = 1) = 1$$

The overall error rate over μ can be calculated as:

$$\begin{aligned} Pr_\mu(h(X, A) \neq Y) &= p * Pr_\mu(h(X, A) \neq Y | A = 0) + (1 - p) * Pr_\mu(h(X, A) \neq Y | A = 1) \\ &= p * Pr_\mu(h(X, A) \neq Y | A = 0) + (1 - p)(1 - Pr_\mu(h(X, A) \neq Y | A = 0)) \end{aligned}$$

It is manifest that the overall error rate is minimal when $Pr_\mu(h(X, A) \neq Y | A = 0) = 0$, which gives the lower bound of the overall error rate:

$$Pr_\mu(h(X, A) \neq Y) \quad \geq \quad 1 - p$$

Q3

Given that a Frobenious norm have following property:

$$\|A\|_{\mathrm{F}} = \sqrt{\sum_{i}^{m}\sum_{j}^{n}|a_{ij}|^2} = \sqrt{\mathrm{trace}\,(A^*A)} = \sqrt{\sum_{i=1}^{\min\{m,n\}}\sigma_i^2(A)},$$

We can use this to express $L(X)$ as:

$$L(X) \quad = \quad \frac{1}{2}tr\left[(P - XX^T)^T(P - XX^T)\right]$$

Given that trace derivation satisfies [1]:

$$\frac{\partial tr[\mathbf{AXB}]}{\partial \mathbf{X}} = \mathbf{A}^T\mathbf{B}^T$$

So, denote the $W = P - XX^T$ taking derivative by chain rule gives:

$$\begin{aligned}\frac{L(X)}{\delta X} &= \frac{L(W)}{\delta W}\frac{L(W)}{\delta X} \\ &= \frac{1}{2}tr[W^TW]\frac{\delta W}{\delta X} \\ &= W\frac{\delta W}{\delta X}\end{aligned}$$

Since $(X^TX)' = 2X$,

$$\frac{\delta W}{\delta x} = -2X$$

Therefore,

$$\begin{aligned}\frac{L(X)}{\delta x} &= W\frac{\delta W}{\delta X} = (P - XX^T)(-2X) \\ &= 2(XX^T - P)X\end{aligned}$$

Q4

4.1 Recall the update step in the gradient descent procedure:

$$w_t = w_{t-1} - \frac{1}{t}\nabla_w \mathcal{L}\left(w_{t-1}\right)$$

By derivation, we can express the gradient as:

$$\nabla_w L(w) = \sum_{i=1}^{n} x_i \left(w \cdot x_i - y_i\right)$$

This gradient can be seen as a linear combination of the input vectors $x_1, x_2, \ldots x_n$. Each update of w using this gradient is therefore in the direction of a linear combination of the x_i vectors.

Since the initial value is w_0 and all subsequent updates are in the span of the x_i vectors, the difference must also be in the span of $\{x_1, x_2, \ldots x_n\}$

4.2.1

Given that the $\mathcal{L}(w)$ have following forms:

$$\begin{aligned} \mathcal{L}(w) &:= \frac{1}{2}\sum_{i=1}^{n}\left(w \cdot x_i - y_i\right)^2 \\ &= \frac{1}{2}\|Xw - y\|_2^2 \end{aligned}$$

So we can also express its gradient in the form of:

$$\nabla\mathcal{L}(w^*) = \left(X^T X\right) w^* - X^T y$$

Not that for such a convex function, the optimal point for linear regression is when the gradient is zero, that is:

$$\left(X^T X\right) w^* - X^T y = 0$$

i.e.

$$X^T X w^* = X^T y$$

4.2.2 To show this we need to use the fact that both w^*, $\hat{w}$ satisfy the normal equation. Thus we have:

$$\begin{aligned} X^T X w^* &= X^T y \\ X^T X \hat{w} &= X^T y \end{aligned}$$

The subtraction of these two equations gives:

$$X^T X(w^* - \hat{w}) = 0$$

Since:

$$X^T X v \Longrightarrow v^T X^T X v = 0 \Longrightarrow \|Xv\|_2^2 = 0 \Longrightarrow Xv = 0$$

We can say that:

$$\begin{aligned} X^T X(w^* - \hat{w}) &= X(w^* - \hat{w}) \\ &= 0 \end{aligned}$$

That is essentially, $X(w^* - \hat{w}) \in Ker(X)$.

4.3

Let's consider a gradient descent algorithm with fixed update step length $0 < \lambda < 1$, for every steps of gradient descent we have:

$$\begin{aligned} \mathrm{w}_k &= \mathrm{w}_{k-1} - \lambda \nabla L(\mathrm{w}_k - 1) \\ &= \left(\mathrm{I} - \lambda \mathrm{X}^{\mathrm{T}}\mathrm{X}\right) \mathrm{w}_k + \lambda \mathrm{X}^{\mathrm{T}}\mathrm{y} \end{aligned}$$

By induction, the above equation simplifies to an expression with w_0:

$$\mathrm{w}_t = \left(\mathrm{I} - \lambda \mathrm{X}^{\mathrm{T}}\mathrm{X}\right)^t \mathrm{w}_0 + \sum_{j=0}^{t-1} \left(\left(\mathrm{I} - \frac{1}{j} \mathrm{X}^{\mathrm{T}}\mathrm{X} \right)^j \right) \frac{1}{k} \mathrm{X}^{\mathrm{T}}\mathrm{y}$$

By SVD decomposition, we have:

$$\mathrm{X} = \mathrm{U}\Sigma\mathrm{V}^T = \mathrm{U} \begin{bmatrix} \Sigma_1 & \mathrm{O} \end{bmatrix} \begin{bmatrix} \mathrm{V}_1^T \\ \mathrm{V}_2^T \end{bmatrix} = \mathrm{U}\Sigma_1\mathrm{V}_1^T$$

By substitution of $X = \mathrm{U}\Sigma\mathrm{V}^T$, we have we the second term be:

$$\sum_{j=0}^{t-1} \left(\left(\mathrm{I} - \lambda \mathrm{X}^{\mathrm{T}}\mathrm{X} \right)^j \right) \frac{1}{t} \mathrm{X}^{\mathrm{T}}\mathrm{y} = V \sum_{j=0}^{t-1} \left(\left(\mathrm{I} - \frac{1}{j} \Sigma^T \Sigma y \right)^j \right) \frac{1}{t} \Sigma^T U^T y$$

Therefore by definition of SVD matrices, we have:

$$V \sum_{j=0}^{t-1} \left(\left(\mathrm{I} - \frac{1}{j} \Sigma^T \Sigma y \right)^j \right) \frac{1}{t} \Sigma^T U^T y = \frac{1}{t} V \Sigma^T \sum_{j=0}^{t-1} \left(\left(\mathrm{I} - \frac{1}{j} \Sigma^T \Sigma y \right)^j \right) U^T y$$

Notice that the part $\left(\mathrm{I} - \lambda \Sigma^T \Sigma\right)$ is actually a diagonal matrix, denote it as P, we have:

$$P_{ii} = \begin{cases} (1 - \lambda\sigma^2)^j : & \sigma_i > 0 \\ 0 : & \sigma_i = 0 \end{cases}$$

Since the absolute value of the term $1 - \lambda\sigma^2$ always less than 1. i.e. $|1 - \lambda\sigma^2| < 1$.
Consider the property of the sum of such infinite series where:

$$\lim_{k \to \infty} \sum_{j=0}^{k} q^k = \frac{1}{1-q}$$

Therefore, assuming the steps needed for gradient descent convergence is extremely large, we have:

$$\lim_{t \to \infty} \sum_{j=0}^{t} P^j = \begin{cases} \frac{1}{(\lambda\sigma^2)} : & \sigma_i > 0 \\ 0 : & \sigma_i = 0 \end{cases}$$

Thus, when $t \to \infty$, denote the result as Q:

$$Q = \lim_{t \to \infty} \lambda \Sigma^T \left(\sum_{j=0}^{t-1} (\mathrm{I} - \lambda \Sigma \Sigma^{\mathrm{T}})^{\mathrm{j}} \right), \quad \text{where } Q_{ii} = \begin{cases} 1/\sigma_i : & \sigma_i > 0 \\ 0 : & \sigma_i = 0 \end{cases}$$

Since Q is obtained by transposing Σ and taking the reciprocal of all its non-zero diagonal entries (singular value), we have $Q = \Sigma^+$ [2]. Substituting $Q = \Sigma^+$ gives:

$$V \lim_{t\to\infty} \lambda\Sigma^T \left(\sum_{j=0}^{t-1} (\mathrm{I} - \lambda\Sigma\Sigma^{\mathrm{T}})^{\mathrm{j}} \right) U^T \longrightarrow V\Sigma^+U^T = X^+$$

Similarly, for the first term, we have:

$$\lim_{t\to\infty} \mathrm{w}_t = \left(\mathrm{I} - \lambda \mathrm{X}^{\mathrm{T}}\mathrm{X}\right)^t \mathrm{w}_0 = 0$$

Here we may have the whole expression for w^* be:

$$w^* = w_{t\to\infty} = X^+y$$

Since X^+ is the Moore-Penrose pseudo inverse of X, therefore X^+y is the solution (if there is one) with the smallest euclidean norm [3] [4].

References

[1] Kaare Brandt Petersen, Michael Syskind Pedersen, et al. The matrix cookbook. *Technical University of Denmark*, 7(15):510, 2008.

[2] Wikipedia contributors. Moore–penrose inverse — Wikipedia, the free encyclopedia, 2023. [Online; accessed 17-September-2023].

[3] Hyperplane (https://math.stackexchange.com/users/99220/hyperplane). Does gradient descent for linear regression select the minimal norm solution? Mathematics Stack Exchange. URL:https://math.stackexchange.com/q/4013058 (version: 2021-02-04).

[4] Rodrigo de Azevedo (https://math.stackexchange.com/users/339790/rodrigo-de azevedo). Does gradient descent converge to a minimum-norm solution in least-squares problems? Mathematics Stack Exchange. URL:https://math.stackexchange.com/q/3499305 (version: 2022-02-18).