Homework 1

2020 Spring CSCI 5525: Machine Learning

Due on Fabruary 16th 11:59pm

Please type in your info:

- **Name**: Pranay Patil
- **Student ID**: 5592264
- **Email**: patil122@umn.edu
- **Collaborators, and on which problems:**

Homework Policy. (1) You are encouraged to collaborate with your classmates on homework problems, but each person must write up the final solutions individually. You need to fill in above to specify which problems were a collaborative effort and with whom. (2) Regarding online resources, you should **not**:

- Google around for solutions to homework problems,
- Ask for help on online.
- Look up things/post on sites like Quora, StackExchange, etc.

Submission. Submit a PDF using this LaTeX template for written assignment part and submit Python jupyter or Colab python notebooks (.ipynb) for all programming part. You should upload all the files on Canvas.

Written Assignment

Instruction. For each problem, you are required to write down a full mathematical proof to establish the claim.

Problem 1. Two helpful matrices.

Let us first recall the notations in linear regression. The design matrix and the response vector are are defined as:

$$A = \begin{bmatrix} \leftarrow x_1^\intercal \rightarrow \\ \vdots \\ \leftarrow x_n^\intercal \rightarrow \end{bmatrix} \qquad \mathbf{b} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

For this problem, we will assume the covariance matrix $A^\intercal A$ is invertible, and so $(A^\intercal A)^{-1}$ is well-defined (**Clearly mention the properties of matrix operations used while solving**).

Problem 1.1. The residual matrix. For any weight vector $\mathbf{w}$, let us define the vector of least squares residuals as

$$e = \mathbf{b} - A\mathbf{w}$$

Now if $\mathbf{w}$ is the least square solution given by $\mathbf{w} = (A^\intercal A)^{-1}A^\intercal\mathbf{b}$, we can rewrite e as

$$e = \mathbf{b} - A(A^\intercal A)^{-1}A^\intercal\mathbf{b} = \left(I - A(A^\intercal A)^{-1}A^\intercal\right)\mathbf{b}$$

Now let $M = \left(I - A(A^\intercal A)^{-1}A^\intercal\right)$. Show that

- M is symmetric (i.e. $M = M^\intercal$). (**2 points**)
- M is idempotent (i.e. $M^2 = M$). (**2 points**)
- $MA = 0$. (**1 point**)

Your answer.

- M is symmetric

  Proof:

  $M = I - A(A^\intercal A)^{-1}A^\intercal$

  $M^\intercal = \left(I - A(A^\intercal A)^{-1}A^\intercal\right)^\intercal$

  $M^\intercal = I^\intercal - \left(A(A^\intercal A)^{-1}A^\intercal\right)^\intercal$

  $M^\intercal = I - \left(A(A^\intercal A)^{-1}A^\intercal\right)^\intercal$

  Using $(XZY)^\intercal = Y^\intercal Z^\intercal X^\intercal$

  $M^\intercal = I - A\left((A^\intercal A)^{-1}\right)^\intercal A^\intercal$

  Using $(X^{-1})^\intercal = (X^\intercal)^{-1}$

  $M^\intercal = I - A\left((A^\intercal A)^\intercal\right)^{-1}A^\intercal$

  $M^\intercal = I - A(A^\intercal A)^{-1}A^\intercal$

  $M^\intercal = M$

- M is idempotent

  Proof:

  $M = I - A(A^\intercal A)^{-1}A^\intercal$

  $M^2 = \left(I - A(A^\intercal A)^{-1}A^\intercal\right)^2$

  Let $H = A(A^\intercal A)^{-1}A^\intercal$

  $M^2 = (I - H)^2$

  $M^2 = (I - H) * (I - H)$

  $M^2 = I(I - H) - H(I - H)$

  $M^2 = I^2 - IH - HI + H^2$

  $M^2 = I^2 - 2H + H^2$

  Now,

$H^2 = \left(A(A^\intercal A)^{-1}A^\intercal\right)^2$

$H^2 = \left(A(A^\intercal A)^{-1}A^\intercal\right) * \left(A(A^\intercal A)^{-1}A^\intercal\right)$

$H^2 = A[(A^\intercal A)^{-1}(A^\intercal A)](A^\intercal A)^{-1}A^\intercal$

$H^2 = AI(A^\intercal A)^{-1}A^\intercal$

$H^2 = A(A^\intercal A)^{-1}A^\intercal$

$H^2 = H$

Therefore,

$M^2 = I^2 - 2H + H$

$M^2 = I^2 - H$

$M^2 = I - A(A^\intercal A)^{-1}A^\intercal$

$M^2 = M$

- $MA = 0$

  Proof:

  $MA = \left(I - A(A^\intercal A)^{-1}A^\intercal\right)A$

  $MA = A - A(A^\intercal A)^{-1}(A^\intercal A)$

  $MA = A - AI$

  $MA = A - A$

  $MA = 0$

Problem 1.2. The hat matrix. Using the residual maker, we can derive another matrix, the hat matrix or projection matrix $P = I - M = A(A^\intercal A)^{-1}A^\intercal$. Note that the predicted value by the least squares solution is given by $P\mathbf{b}$. Show that

- P is symmetric. (**1 point**)
- P is idempotent. (**1 point**)

Your answer.

- P is symmetric

  Proof:

  $P = A(A^\intercal A)^{-1}A^\intercal$
  $P^\intercal = \left(A(A^\intercal A)^{-1}A^\intercal\right)^\intercal$
  $P^\intercal = A\left((A^\intercal A)^{-1}\right)^\intercal A^\intercal$
  $P^\intercal = A\left((A^\intercal A)^\intercal\right)^{-1}A^\intercal$
  $P^\intercal = A(A^\intercal A)^{-1}A^\intercal$
  $P^\intercal = P$

- P is idempotent

  Proof:

$$
\begin{aligned}
P &= A(A^\intercal A)^{-1}A^\intercal \\
P^2 &= \left(A(A^\intercal A)^{-1}A^\intercal\right)^2 \\
P^2 &= \left(A(A^\intercal A)^{-1}A^\intercal\right) * \left(A(A^\intercal A)^{-1}A^\intercal\right) \\
P^2 &= A(A^\intercal A)^{-1}(A^\intercal A)(A^\intercal A)^{-1}A^\intercal \\
P^2 &= AI(A^\intercal A)^{-1}A^\intercal \\
P^2 &= A(A^\intercal A)^{-1}A^\intercal \\
P^2 &= P
\end{aligned}
$$

Problem 2. Gradient of conditional log-likelihood.

For any $a \in \mathbb{R}$, let $\sigma(a) = \frac{1}{1+\exp(-a)}$. For each example $(x_i, y_i) \in \mathbb{R}^d \times \{0, 1\}$, the conditional log-likelihood of logistic regression is

$$\ell(y_i \mid x_i, \mathbf{w}) = y_i \ln(\sigma(\mathbf{w}^\intercal x_i)) + (1 - y_i) \ln(\sigma(-\mathbf{w}^\intercal x_i))$$

Derive the gradient of $\ell(y_i \mid x_i, \mathbf{w})$ with respect to w_j (i.e. the j-th coordinate of $\mathbf{w}$), i.e. $\frac{\partial}{\partial w_j}\ell(y_i \mid x_i, \mathbf{w})$ (**Clearly mention the properties of derivatives used while solving**). (**6 points**)

Your answer. First consider the function $\sigma(a)$

$$
\begin{aligned}
&\sigma(a) = \frac{1}{1+\exp(-a)} \\
&\textit{Taking derivative w.r.t. } a \\
&\frac{\partial}{\partial a}\sigma(a) = \frac{\partial}{\partial a}[\frac{1}{1+\exp(-a)}] \\
&\textit{By chain rule} \\
&\frac{\partial}{\partial a}\sigma(a) = -1 * \frac{1}{(1+\exp(-a))^2} * \frac{\partial}{\partial a}\exp(-a) \\
&\frac{\partial}{\partial a}\sigma(a) = -1 * \frac{1}{(1+\exp(-a))^2} * \exp(-a) * -1 \\
&\frac{\partial}{\partial a}\sigma(a) = \frac{\exp(-a)}{(1+\exp(-a))^2} \\
&\frac{\partial}{\partial a}\sigma(a) = \frac{1+\exp(-a)-1}{(1+\exp(-a))^2} \\
&\frac{\partial}{\partial a}\sigma(a) = \frac{1+\exp(-a)}{(1+\exp(-a))^2} - \frac{1}{(1+\exp(-a))^2} \\
&\frac{\partial}{\partial a}\sigma(a) = \frac{1}{(1+\exp(-a))} - \frac{1}{(1+\exp(-a))^2)} \\
&\frac{\partial}{\partial a}\sigma(a) = \frac{1}{1+\exp(-a)}[1 - \frac{1}{(1+\exp(-a)}]
\end{aligned}
$$

$$\frac{\partial}{\partial a}\sigma(a) = \sigma(a)(1 - \sigma(a)) \tag{1}$$

Now consider the conditional log likelihood:

$$l(y_i \mid x_i, w) = y_i \ln(\sigma(w^\intercal x_i)) + (1 - y_i) \ln(\sigma(-w^\intercal x_i))$$

Taking derivative w.r.t. w_j

$$\frac{\partial}{\partial w_j} l(y_i \mid x_i, w) = \frac{\partial}{\partial w_j} [y_i \ln(\sigma(w^\intercal x_i)) + (1 - y_i) \ln(\sigma(-w^\intercal x_i))]$$

Distributing differential

$$\frac{\partial}{\partial w_j} l(y_i \mid x_i, w) = \frac{\partial}{\partial w_j} [y_i \ln(\sigma(w^\intercal x_i))] + \frac{\partial}{\partial w_j} [(1 - y_i) \ln(\sigma(-w^\intercal x_i))]$$

$$\frac{\partial}{\partial w_j} l(y_i \mid x_i, w) = y_i * \frac{\partial}{\partial w_j} [\ln(\sigma(w^\intercal x_i))] + (1 - y_i) * \frac{\partial}{\partial w_j} [\ln(\sigma(-w^\intercal x_i))]$$

But $\sigma(-a) = 1 - \sigma(a)$

$$\frac{\partial}{\partial w_j} l(y_i \mid x_i, w) = y_i * \frac{\partial}{\partial w_j} [\ln(\sigma(w^\intercal x_i))] + (1 - y_i) * \frac{\partial}{\partial w_j} [\ln(1 - \sigma(w^\intercal x_i))]$$

By chain rule

$$\frac{\partial}{\partial w_j} l(y_i \mid x_i, w) = y_i * \frac{1}{\sigma(w^\intercal x_i)} \frac{\partial}{\partial w_j} \sigma(w^\intercal x_i) + (1 - y_i) * \frac{1}{1 - \sigma(w^\intercal x_i)} \frac{\partial}{\partial w_j} (1 - \sigma(w^\intercal x_i))$$

$$\frac{\partial}{\partial w_j} l(y_i \mid x_i, w) = y_i * \frac{1}{\sigma(w^\intercal x_i)} \frac{\partial}{\partial w_j} \sigma(w^\intercal x_i) - (1 - y_i) * \frac{1}{1 - \sigma(w^\intercal x_i)} \frac{\partial}{\partial w_j} \sigma(w^\intercal x_i)$$

$$\frac{\partial}{\partial w_j} l(y_i \mid x_i, w) = \frac{\partial}{\partial w_j} \sigma(w^\intercal x_i)(y_i * \frac{1}{\sigma(w^\intercal x_i)} - (1 - y_i) * \frac{1}{1 - \sigma(w^\intercal x_i)})$$

From (1) *and chain rule*

$$\frac{\partial}{\partial w_j} l(y_i \mid x_i, w) = \sigma(w^\intercal x_i)(1 - \sigma(w^\intercal x_i)) \frac{\partial}{\partial w_j} w^\intercal x_i (y_i * \frac{1}{\sigma(w^\intercal x_i)} - (1 - y_i) * \frac{1}{1 - \sigma(w^\intercal x_i)})$$

$$\frac{\partial}{\partial w_j} l(y_i \mid x_i, w) = \frac{\partial}{\partial w_j} w^\intercal x_i (y_i * \frac{\sigma(w^\intercal x_i)(1 - \sigma(w^\intercal x_i))}{\sigma(w^\intercal x_i)} - (1 - y_i) * \frac{\sigma(w^\intercal x_i)(1 - \sigma(w^\intercal x_i))}{1 - \sigma(w^\intercal x_i)})$$

$$\frac{\partial}{\partial w_j} l(y_i \mid x_i, w) = \frac{\partial}{\partial w_j} w^\intercal x_i (y_i(1 - \sigma(w^\intercal x_i)) - (1 - y_i)\sigma(w^\intercal x_i))$$

$$\frac{\partial}{\partial w_j} l(y_i \mid x_i, w) = \frac{\partial}{\partial w_j} w^\intercal x_i (y_i - y_i * \sigma(w^\intercal x_i) - \sigma(w^\intercal x_i) + y_i * \sigma(w^\intercal x_i))$$

$$\frac{\partial}{\partial w_j} l(y_i \mid x_i, w) = \frac{\partial}{\partial w_j} w^\intercal x_i (y_i - \sigma(w^\intercal x_i))$$

Now $w^\intercal x_i = w_1 x_{i1} + w_2 x_{i2} + ... + w_n x_{in}$,

so $\frac{\partial}{\partial w_j} w^\intercal x_i = 0 + 0 + ... \frac{\partial}{\partial w_j} w_j x_{ij} ... + 0$

$$\frac{\partial}{\partial w_j} l(y_i \mid x_i, w) = x_{ij}(y_i - \sigma(w^\intercal x_i))$$

Problem 3. Derivation of Ridge Regression Solution.

Recall that in class we claim that the solution to ridge regression ERM:

$$\min_{\mathbf{w}} \ (\|A\mathbf{w} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{w}\|_2^2)$$

is $\mathbf{w}^* = (A^\intercal A + \lambda I)^{-1} A^\intercal \mathbf{b}$. Now provide a proof. (**6 points**)
(Hint: recall that $\nabla F(\mathbf{w}) = \mathbf{0}$ is a sufficient condition for $\mathbf{w}$ to be a minimizer of any convex function F. To get a full credit, you should be able to show why $A^\intercal A + \lambda I$ is invertible.)
(**Clearly mention the properties of matrix calculus used while solving**)

Your answer.

$$F(w) = \min_w(\|Aw - b\|_2^2 + \lambda\|w\|_2^2)$$
$$Differentiating\ F(w)\ wrt\ w$$
$$\nabla F(\mathbf{w}) = \frac{\partial}{\partial w} F(w)$$
$$\nabla F(\mathbf{w}) = \frac{\partial}{\partial w}(\|Aw - b\|_2^2 + \lambda\|w\|_2^2)$$
$$\nabla F(\mathbf{w}) = \frac{\partial}{\partial w}\|Aw - b\|_2^2 + \frac{\partial}{\partial w}\lambda\|w\|_2^2$$
$$\nabla F(\mathbf{w}) = 2(Aw - b)\frac{\partial}{\partial w}Aw + 2\lambda w$$
$$\nabla F(\mathbf{w}) = 2A^\intercal(Aw - b) + 2\lambda w$$
$$To\ minimize\ F(w)\ we\ will\ equate\ \nabla F(\mathbf{w}^*)\ to\ 0$$
$$2A^\intercal(Aw^* - b) + 2\lambda w^* = 0$$
$$A^\intercal(Aw^* - b) + \lambda w^* = 0$$
$$A^\intercal Aw^* - A^\intercal b + \lambda w^* = 0$$
$$A^\intercal Aw^* + \lambda w^* = A^\intercal b$$
$$w^*(A^\intercal A + \lambda I) = A^\intercal b$$
$$w^* = (A^\intercal A + \lambda I)^{-1} A^\intercal b$$

Here, since we are adding a positive constant λ to the diagonal of $A^\intercal A$, $A^\intercal A + \lambda I$ it will always be invertible, even for the cases where $A^\intercal A$ is singular.
To prove that w^* is the optimal minimizer, we can verify if the 2^{nd} derivative of $F(w)$ is positive:

$$From\ above\ calculations$$
$$\nabla^2 F(\mathbf{w}) = \frac{\partial}{\partial w}(2A^\intercal Aw + 2\lambda w - 2A^\intercal b)$$
$$\nabla^2 F(\mathbf{w}) = \frac{\partial}{\partial w}2A^\intercal Aw + \frac{\partial}{\partial w}2\lambda w - \frac{\partial}{\partial w}2A^\intercal b$$
$$\nabla^2 F(\mathbf{w}) = 2A^\intercal A + 2\lambda$$

It can be seen that $A^\intercal A$ is just a square of values in A and λ is a positive constant. Hence $\nabla^2 F(\mathbf{w}) > 0$ and w^* is an optimal solution for $\min_w(\|Aw - b\|_2^2 + \lambda\|w\|_2^2)$

Problem 4. Minimizing a Squared Norm Plus an Affine Function.

A generalization of the least squares problem adds an affine function to the least squares objective,

$$\min_{\mathbf{w}} \ \|A\mathbf{w} - \mathbf{b}\|_2^2 + c^\intercal \mathbf{w} + d$$

where $A \in \mathbb{R}^{m\times n}, \mathbf{w} \in \mathbb{R}^n, \mathbf{b} \in \mathbb{R}^m, c \in \mathbb{R}^n, d \in \mathbb{R}$. Assume the column of A are linearly independent. This generalized problem can be solved by reducing it to a standard least squares problem, using a trick called *completing the square*.

Show that the objective of the problem above can be expressed in the form

$$\|A\mathbf{w} - \mathbf{b}\|_2^2 + c^\intercal \mathbf{w} + d = \|A\mathbf{w} - \mathbf{b} + \mathbf{f}\|_2^2 + g$$

where $\mathbf{f} \in \mathbb{R}^m, g \in \mathbb{R}$.It follows that we can solve the generalized least squares problem by minimizing $\|A\mathbf{w} - (\mathbf{b} - \mathbf{f})\|_2^2$

(Hint: Express the norm squared term on the right-hand side as $\|(A\mathbf{w} - \mathbf{b}) + \mathbf{f})\|_2^2$ and expand it. Then argue that the equality above holds provided $2A^\intercal \mathbf{f} = c$. One possible choice is $f = \frac{1}{2}(A^\dagger)^\intercal c$.) (You must justify these statements.) (**6 point**)

Your answer. To prove the first part we will start from the RHS:

$$\|Aw - b + f\|_2^2 + g = \|(Aw - b) + f\|_2^2 + g$$
$$\|Aw - b + f\|_2^2 + g = (Aw - b)^\intercal (Aw - b) + 2f^\intercal (Aw - b) + f^\intercal f + g$$
$$\|Aw - b + f\|_2^2 + g = \|(Aw - b) + f\|_2^2 + 2f^\intercal Aw - 2f^\intercal b + f^\intercal f + g$$
$$\|Aw - b + f\|_2^2 + g = \|(Aw - b) + f\|_2^2 + 2f^\intercal Aw + f^\intercal (f - 2b) + g$$

Comparing '$\|(Aw - b) + f\|_2^2 + 2f^\intercal Aw + f^\intercal (f - 2b) + g$' to '$\|A - b\|_2^2 + c^\intercal w + d$' we get:

$$c^\intercal w = 2f^\intercal Aw$$
$$d = f^\intercal (f - 2b) + g$$

For the second part:
We know that

$$2f^\intercal Aw = c^\intercal w$$
$$2f^\intercal Aw - c^\intercal w = 0$$
$$2(A^\intercal f)^\intercal w - c^\intercal w = 0$$
$$(2(A^\intercal f)^\intercal - c^\intercal)w = 0$$

This holds if $2(A^\intercal f)^\intercal = c^\intercal$ or simply $2A^\intercal f = c$
To justify the possible value of f given in the hint, lets substitute it in the above equation

$$
\begin{aligned}
2A^\intercal f &= 2A^\intercal \frac{1}{2}(A^\dagger)^\intercal c \\
Now, A^\dagger &= (A^\intercal A)^{-1}A^\intercal \\
2A^\intercal f &= 2A^\intercal \frac{1}{2}((A^\intercal A)^{-1}A^\intercal)^\intercal c \\
2A^\intercal f &= A^\intercal(A((A^\intercal A)^{-1})^\intercal)c \\
2A^\intercal f &= A^\intercal(A((A^\intercal A)^\intercal)^{-1})c \\
2A^\intercal f &= A^\intercal A((A^\intercal A)^\intercal)^{-1}c \\
2A^\intercal f &= A^\intercal A(A^\intercal A)^{-1}c \\
2A^\intercal f &= (A^\intercal A)(A^\intercal A)^{-1}c \\
2A^\intercal f &= Ic \\
2A^\intercal f &= c
\end{aligned}
$$

Problem 5. Iterative Method for Least Squares Problem.

In this exercise we explore an iterative method, due to the mathematician Lewis Richardson, that can be used to compute $\hat{\mathbf{w}} = A^\dagger \mathbf{b}$, we define $\mathbf{w}^{(1)} = 0$ and for $k = 1, 2, 3, ...,$

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \mu A^\intercal(A\mathbf{w}^{(k)} - \mathbf{b})$$

where μ is a positive parameter, and the superscripts denote the iteration number.This defines a sequence of vectors that converge to $\hat{\mathbf{w}}$ provided μ is not too large; The iteration is terminated when $A^\intercal(A\mathbf{w}^{(k)} - \mathbf{b})$ is small enough, which means the least squares optimality conditions are almost satisfied. To implement the method we only need to multiply vectors by A and by $A^\intercal$. If we have efficient methods for carrying out these two matrix-vector multiplications, this iterative method can be faster. Iterative methods are often used for very large scale least squares problems.

(a) Show that if $\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)}$, we have $\mathbf{w}^{(k)} = \hat{\mathbf{w}}$. (**5 points**)

(b) Express the vector sequence $x^{(k)}$ as a linear dynamical system with constant dynamics matrix and offset, i.e., in the form $\mathbf{w}^{(k+1)} = F\mathbf{w}^{(k)} + g$. (**3 points**)

Your answer (a) Solution to the least square solutions is given by,

$f(w) = \|Aw - b\|_2^2$

Differentiating wrt to w get the optimal solution

$\nabla f(w) = \dfrac{\partial}{\partial w}\|Aw - b\|_2^2$

$\nabla f(w) = 2A^\intercal(Aw - b)$

If $\hat{w}$ is the optimum solution, then $\nabla f(w) = 0$

$2A^\intercal(A\hat{w} - b) = 0$

$A^\intercal(A\hat{w} - b) = 0$

This is first order condition

Now, consider the given iterative solution:

$w^{(k+1)} = w^{(k)} - \mu A^\intercal (Aw^{(k)} - b)$
$If\ w^{(k+1)} = w^{(k)},$
$w^{(k)} = w^{(k)} - \mu A^\intercal (Aw^{(k)} - b)$
$\mu A^\intercal (Aw^{(k)} - b) = w^{(k)} - w^{(k)}$
$\mu A^\intercal (Aw^{(k)} - b) = 0$
$A^\intercal (Aw^{(k)} - b) = 0$
$From\ first\ order\ condition,\ w^{(k)} is\ the\ optimal\ solution$
$w^{(k)} = \hat{w}$

Your answer (b)

$w^{(k+1)} = w^{(k)} - \mu A^\intercal (Aw^{(k)} - b)$
$w^{(k+1)} = w^{(k)} - \mu A^\intercal A w^{(k)} + \mu A^\intercal b$
$w^{(k+1)} = (I - \mu A^\intercal A) w^{(k)} + (\mu A^\intercal b)$
$w^{(k+1)} = F w^{(k)} + g$
$where,$
$F = constant\ dynamics\ matrix = I - \mu A^\intercal A$
$g = offset = \mu A^\intercal b$

Programming Assignment

Instruction. For each problem, you are required to report descriptions and results in the PDF and submit code as python file (.py) (as per the question).

- **Python** version: Python 3.
- Please follow PEP 8 style of writing your Python code for better readability in case you are wondering how to name functions & variables, put comments and indent your code
- **Packages allowed**: numpy, pandas, matplotlib
- Please PROPERLY COMMENT your code in order to have utmost readability
- Please provide the required functions for each problem.
- There shall be NO runtime errors involved.
- There would be PENALTY if any of the above is not followed
- **Submission**: For programming parts, **ONLY THE PYTHON 3 NOTEBOOKS WILL BE ACCEPTED**

Problem 6. Iterative Method for Least Squares Problem (Cont'd).

For this problem, you will implement Richardson algorithm introduced in Problem 5 and report the required graphs. Please submit a Python script file (name hw1_lsq_iter.py)

- Generate a random 20×10 matrix A and 20-vector b, and compute $\hat{\mathbf{w}} = A^{\dagger}\mathbf{b}$. Run the Richardson algorithm with $\mu = \frac{1}{\|A\|^2}$ for 500 iterations, and plot $\|\mathbf{w}^{(k)} - \hat{\mathbf{w}}\|$ to verify that $x^{(k)}$ appears to be converging to $\hat{x}$. Please properly analyze and describe your plot. (**12 points**)
- Note: function np.linalg.lstsq is not allowed.

  To get a full credit, the main script should output the required plots with explicitly named x-y axis and the following function should be provided:

  1. $\mathbf{w}$ = lsq(A, $\mathbf{b}$) where $\mathbf{w} = \arg\min_{\mathbf{w}} \ \|A\mathbf{w} - \mathbf{b}\|_2^2$ solved by closed form.
  2. $\mathbf{w}$ = lsq_iter(A, $\mathbf{b}$) where $\mathbf{w} = \arg\min_{\mathbf{w}} \ \|A\mathbf{w} - \mathbf{b}\|_2^2$ solved by Richardson algorithm.

Your answer. As you can see in the figure 1, the value of w is getting closer and closer to the optimal value of $\hat{w}$. After about 110 iterations, $w^{(k)}$ is converged with $\hat{w}$.

Problem 7. Logistic regression.

For this problem, you will use the IRIS dataset. Features are in the file IRISFeat.csv and labels in the file IRISlabel.csv. The dataset has 150 samples. A python script (name hw1-logistic.py) with all the steps need to be submitted.

a) (**12 Points**) Your goal is to implement logistic regression. You can design or structure your code to fulfil the requirements but to get a full credit, the main script should output the required tables or plots with explicitly named x-y axis and the following function should be provided where X being features and y target.:

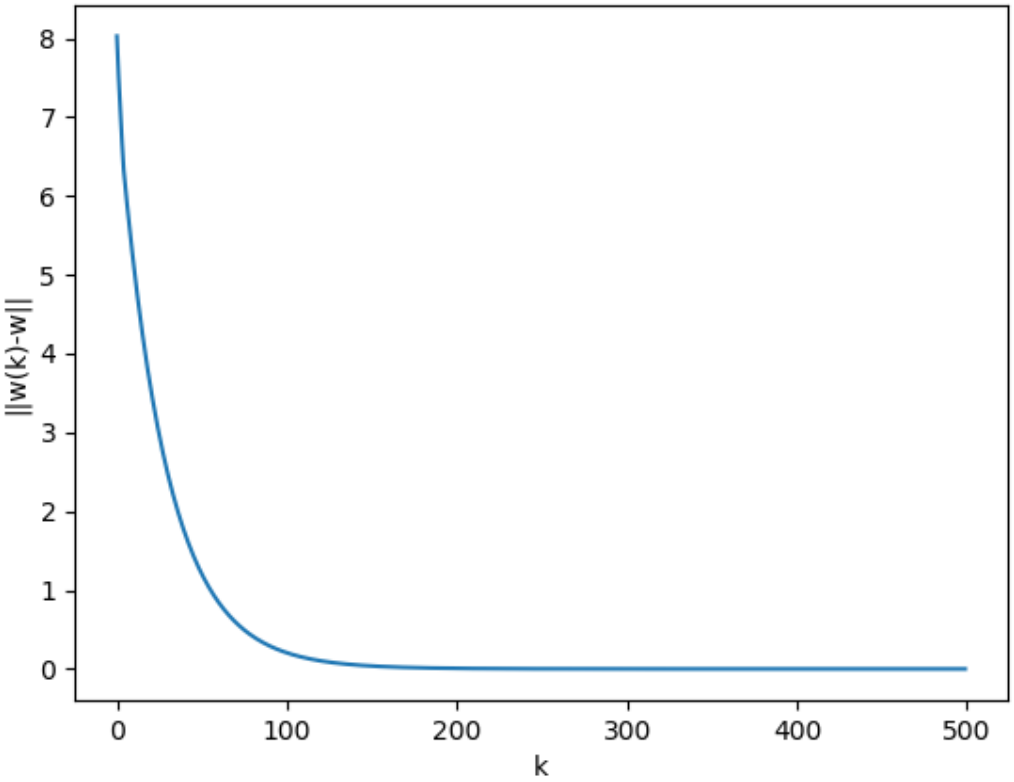


Figure 1: $\|w^{(k)} - \hat{w}\|$

1. Cross validation: Make sure you randomly shuffle the dataset and partition it into almost equal (k=5) folds. Save each of the 5 folds into dictionary X_shuffled and y_shuffled.
   X_train, y_train, X_valid, y_valid = get_next_train_valid(X_shuffled, y_shuffled, itr) where itr is iteration number.
2. model_weights, model_intercept = train(X_train, y_train)
3. y_predict_class = predict(X_valid, model_weights, model_intercept)

   USE GRADIENT DESCENT to solve it. You should initialize the weights randomly to begin with.

b) (**3 Points**) At the beginning, briefly describe the approach in one paragraph along with any equations and methods used in your implementation.

c) (**5 Points**) Report the plot of training and validation set error rates (number of misclassified samples/total number of samples) and the confusion matrix for validation set from 5-fold cross validation. Explain your selection of learning rate and How does it affect the performance/training of your model?

Your answer. b) Following approach is used for training a logistic regression model It has following functions:

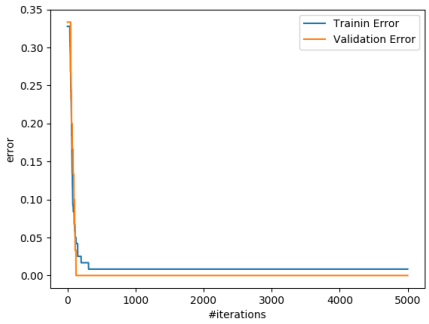
1. **shuffle_data_in_k_folds:** It shuffles the features and labels data and divide it in k partitions.
2. **get_next_train_valid:** It return the training and validation sets for a given iteration.
3. **train:** This method runs gradient descent algorithm for maximum of 1000 iterations or until the $\|dw\|_2^2 >= 0.001$ with learning rate of 0.1. model_weights (W) and model_intercept are updated using following equations

   a) $dw = X^\intercal(y_i - \frac{1}{1+e^{-(w^\intercal X_i+b)}})$
   b) $db = y_i - \frac{1}{1+e^{-(w^\intercal X_i+b)}}$
   c) $w = w - \frac{1}{N} * learning\_rate * dw$
   d) $b = b - \frac{1}{N} * learning\_rate * db$
4. **predict:** It uses following equation to predict the class given X_i
   $y = sign(\frac{1}{1+e^{-(w^\intercal X_i+b)}})$

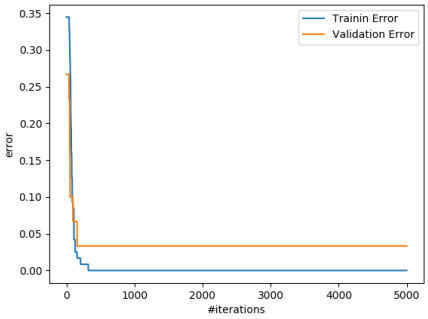
Your answer. c) Training and validation plots

1. Plot of error rates for gradient descent:
2. Confusion matrices for 5 iterations:

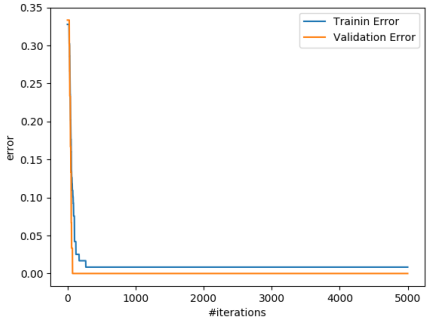
Number of iterations and learning rate were decided after running train method over a range of values. Tried learning rate in the range of (0.01 to 0.9), got best results at 0.1. Number of iterations were tested for the range(1000 to 5000). 5000 gave higher accuracy across all folds and also helped in converging to the optimal solution since our learning rate is relatively lower.



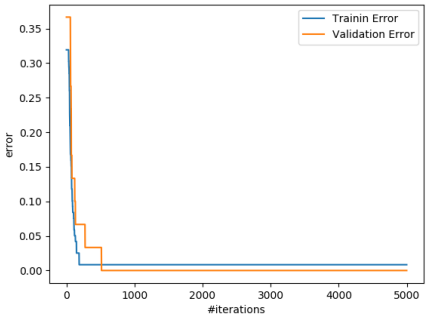
(a) Fold 1



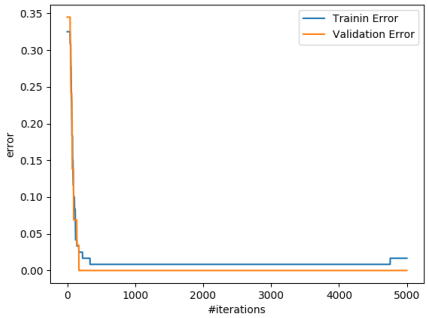
(b) Fold 2



(c) Fold 3



(d) Fold 4



(e) Fold 5

Figure 2: Training and validation error rates

Lower learning rate requires more iterations to converge, but higher learning rate even though takes less iterations, posses the risk of jumping over the optimum solution. Also low learning rate takes more time and high learning rate takes less time.

Confusion matrix #iteration 1

Classes/Prediction	0	1
0	10	0
1	0	20

(a) Fold 1

Confusion matrix #iteration 2

Classes/Prediction	0	1
0	7	0
1	1	22

(b) Fold 2

Confusion matrix #iteration 3

Classes/Prediction	0	1
0	10	0
1	0	20

(c) Fold 3

Confusion matrix #iteration 4

Classes/Prediction	0	1
0	11	0
1	0	19

(d) Fold 4

Confusion matrix #iteration 5

Classes/Prediction	0	1
0	10	0
1	0	19

(e) Fold 5

Figure 3: Confusion matrix for validation set