CS484 - Introduction to Machine Learning
Assignment - 1

Problem - 1 :-

1) Given,
$Q \in R^{N \times N}$, $P \in R^{M \times M}$, and $B \in R^{M \times N}$

we must verify the following identity:

$$(Q^{-1} + B^T P^{-1} B)^{-1} B^T P^{-1} = Q B^T (B Q B^T + P)^{-1}$$

Here, we use Sherman - Morrison - Woodbury formula to prove

$$(A+UCV)^{-1} = A^{-1} - A^{-1} U (C C^{-1} + V A^{-1} U)^{-1} V A^{-1}$$

Here,
A is invertible matrix
U and V are matrices
C is an invertible matrix

So, $A = Q^{-1}$
$U = B^T$
$C = P^{-1}$
$V = B$

Using these our LHS becomes :-

$$(Q^{-1} + B^T P^{-1} B)^{-1}$$

And the RHS becomes :-

$$Q - Q B^T (P + B Q B^T)^{-1} B Q$$

Now to simply let us multiply both side with $(B^T P^{-1})$ to the equation. we obtain

$$(Q^{-1} + B^T P^{-1} B)^{-1} B^T P^{-1} = Q B^T (P + B Q B^T)^{-1}$$

So, As the matrix multiplication is associative we can rewrite the equation as:

$$Q B^T (P + B Q B^T)^{-1} = Q B^T (B Q B^T + P)^{-1}$$

Hence, the identity is proved.

2) Verify woodbury Identity

$$(A+BD^{-1}C)^{-1} = A^{-1} - A^{-1}B(D+CA^{-1}B)^{-1}CA^{-1}$$

Step 1:- Multiply both sides with $(A+BD^{-1}C)$

$$(A+BD^{-1}C)^{-1}(A+BD^{-1}C) = (A^{-1} - A^{-1}B(D+CA^{-1}B)^{-1}CA^{-1})(A+BD^{-1}C)$$

Step 2:- LHS becomes I, as we know

$$(A+BD^{-1}C)^{-1}(A+BD^{-1}C) = I$$

$$I = (A^{-1} - A^{-1}B(D+CA^{-1}B)^{-1}CA^{-1})(A+BD^{-1}C)$$

Step 3:- Now let us solve the RHS, we get

$$A^{-1}A + A^{-1}BD^{-1}C - A^{-1}B(D+CA^{-1}B)^{-1}CA^{-1}A - A^{-1}B(D+CA^{-1}B)^{-1}CA^{-1}BD^{-1}C$$

Step 4:- After simplifying RHS, we get

$$I + A^{-1}BD^{-1}C - A^{-1}B(D+CA^{-1}B)^{-1}C - A^{-1}B(D+CA^{-1}B)^{-1}CA^{-1}BD^{-1}C$$

Step 5:- Let us simplify further, we get

$$(D+CA^{-1}+B)^{-1}(D+CA^{-1}+B) = I$$

So, we get

$$I + A^{-1}BD^{-1}C - A^{-1}B^{-1}C$$

Step 6 :- Now combine LHS & RHS

$$I = I + A^{-1}BD^{-1}C - A^{-1}BD^{-1}C$$

$$I = I + 0$$

$$I = I$$

Hence, we get LHS = RHS.

So, Proved.

Problem - 2 :-

1) Given,

$$x = [x_1 ; x_2 ; x_3] \in R^3$$

$$y = [y_1 ; y_2] \in R^2$$

$$y_1 = x_1^2 - x_2$$

$$y_2 = x_3^2 + 3x_2$$

Now, we need to compute Jacobian matrix $\frac{\partial y}{\partial x}$

The Jacobian matrix is :

$$\frac{\partial y}{\partial x} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \frac{\partial y_1}{\partial x_3} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \frac{\partial y_2}{\partial x_3} \end{bmatrix}$$

Partial Derivatives :-

$$\frac{\partial y_1}{\partial x_1} = \frac{\partial}{\partial x_1}(x_1^2 - x_2) = 2x_1 \qquad \frac{\partial y_2}{\partial x_1} = \frac{\partial}{\partial x_1}(x_3^2 + 3x_2) = 0$$

$$\frac{\partial y_1}{\partial x_2} = \frac{\partial}{\partial x_2}(x_1^2 - x_2) = -1 \qquad \frac{\partial y_2}{\partial x_2} = \frac{\partial}{\partial x_2}(x_3^2 + 3x_2) = 3$$

$$\frac{\partial y_1}{\partial x_3} = \frac{\partial}{\partial x_3}(x_1^2 - x_2) = 0 \qquad \frac{\partial y_2}{\partial x_3} = \frac{\partial}{\partial x_3}(x_3^2 + 3x_2) = 2x_3$$

$$\frac{\partial y}{\partial x} = \begin{bmatrix} 2x_1 & -1 & 0 \\ 0 & 3 & 2x_3 \end{bmatrix}$$

2) Given,

$$x = r \sin\theta \cos\phi$$
$$y = r \sin\theta \sin\phi$$
$$z = r \cos\theta$$

where $r > 0$, $0 < \theta < \pi$, and $0 \leq \phi < 2\pi$

The Jacobian matrix is:

$$\frac{\partial x}{\partial y} = \frac{\partial [x; y; z]}{\partial [r; \theta; \phi]}$$

$$= \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial \theta} & \frac{\partial x}{\partial \phi} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial \theta} & \frac{\partial y}{\partial \phi} \\ \frac{\partial z}{\partial r} & \frac{\partial z}{\partial \theta} & \frac{\partial z}{\partial \phi} \end{bmatrix}$$

Now, we need to calculate the Partial Derivatives of x, y, z

$$x = r \sin\theta \cos\phi$$

$$\frac{\partial x}{\partial r} = \frac{\partial}{\partial r}(r \sin\theta \cos\phi) = \sin\theta \cos\phi$$

$$\frac{\partial x}{\partial \theta} = \frac{\partial}{\partial \theta}(r \sin\theta \cos\phi) = r \cos\theta \cos\phi$$

$$\frac{\partial x}{\partial \phi} = \frac{\partial}{\partial \phi}(r \sin\theta \cos\phi) = -r \sin\theta \sin\phi$$

$$y = r \sin\theta \sin\phi$$

$$\frac{\partial y}{\partial r} = \frac{\partial}{\partial r}(r \sin\theta \sin\phi) = \sin\theta \sin\phi$$

$$\frac{\partial y}{\partial \theta} = \frac{\partial}{\partial \theta}(r \sin\theta \sin\phi) = r \cos\theta \sin\phi$$

$$\frac{\partial y}{\partial \phi} = \frac{\partial}{\partial \phi}(r \sin\theta \sin\phi) = r \sin\theta \cos\phi$$

$$z = r\cos\theta$$

$$\frac{\partial z}{\partial r} = \frac{\partial}{\partial r}(r\cos\theta) = \cos\theta$$

$$\frac{\partial z}{\partial \theta} = \frac{\partial}{\partial \theta}(r\cos\theta) = -r\sin\theta$$

$$\frac{\partial z}{\partial \phi} = \frac{\partial}{\partial \phi}(r\cos\theta) = 0$$

The Jacobian matrix is:

$$\frac{\partial x}{\partial y} = \begin{bmatrix} \sin\theta\cos\phi & r\cos\theta\cos\phi & -r\sin\theta\sin\phi \\ \sin\theta\sin\phi & r\cos\theta\sin\phi & r\sin\theta\cos\phi \\ \cos\theta & -r\sin\theta & 0 \end{bmatrix}$$

Problem - 3 :-

1) Given,

$$L(w) = \frac{1}{2} \sum_{i=1}^{n} (x_i^T w - y_i)^2$$

Here, we need to find Hessian of Least Square Loss

Step 1 :- Express the loss function in matrix form

Loss function can be written as:

$$L(w) = \frac{1}{2} \|Xw - y\|^2$$

So, we can write the above loss function as

$$L(w) = \frac{1}{2} (Xw - y)^T (Xw - y)$$

Step 2 :- compute the gradient of $L(w)$ with respect to w.

we get;

$$\frac{\partial}{\partial w} L(w) = X^T (Xw - y)$$

Step 3 :- Now, we compute Hessian, which is the derivative of gradient

$$\frac{\partial^2}{\partial w^2} L(w) = \frac{\partial}{\partial w} (X^T (Xw - y)) = X^T X$$

So, Hessian is $X^T X$

2) Given,

$$w^* = (XX^T)^{-1} Xy$$

Now, we need to show that the first iteration of Newton's method gives us the above equation.

Newton's method is an iterative optimization algorithm. It updates parameters w according to the rule:

$$w_{new} = w - H(w)^{-1} \nabla_w L(w)$$

where,

* w_{new} is the updated parameter vector
* $H(w)$ is the Hessian matrix
* $\nabla_w L(w)$ is the gradient of loss function

Step 1:- Substitute the values of $H(w)$ and $\nabla_w L(w)$. From previous problem we know that,

$$H(w) = X^T X$$

$$\nabla_w L(w) = X^T(Xw - y)$$

$$w_{new} = w - (X^T X)^{-1} X^T (Xw - y)$$

Step 2:- Now, we need to simplify the expression above

$$w_{new} = w - (X^T X)^{-1} X^T X w + (X^T X)^{-1} X^T y$$

Here, $(X^T X)^{-1} X^T X w = w$ because $(X^T X)^{-1} X^T X$ is an identity matrix.

So,

$$w_{new} = w - w + (X^T X)^{-1} X^T y$$

$$w_{new} = (X^T X)^{-1} X^T y$$

The w_{new} is the solution to the normal equations for the least squares problem:

So,

$$w^* = (X^T X)^{-1} X^T y$$

This is the optimal solution to the least squares problem, which means that Newton's method converges to the optimal solution just after one iteration.

Problem - 4 :-

1) Here, we are asked to show that minimizing the least square loss subject to an l_p-norm constraint is equivalent to minimizing the regularized least square loss with an l_p-norm regularization term.

Given,

$$\min_{w} \sum_{n=1}^{N} (f(x_n; w) - t_n)^2 \quad s.t. \quad \|w\|_p \leq r$$

Here,

* w is the model parameters (weights)
* $f(x_n; w)$ is the model prediction for data point x_n
* t_n is the target value for data point x_n.
* $\|w\|_p$ is the l_p-norm of the vector w, which can represent l_1-norm and l_2-norm depending on the choice of p.

we need to show that this constrained optimization is equivalent to the below unconstrained optimization problem:

$$\min_{w} \sum_{n=1}^{N} (f(x_n; w) - t_n)^2 + \lambda \|w\|_p^p$$

where λ is regulation parameter.

Step 1:- Form the lagrangian.

To solve the constrained optimization problem, we use lagrange multipliers. The idea is to transform the constrained into unconstrained one by introducing a 'lagrange multiplier' λ.

The lagrangian for the constrained is:-

$$\mathcal{L}(w, \lambda) = \sum_{n=1}^{N} (f(x_n; w) - t_n)^2 + \lambda(\|w\|_p^p - \gamma)$$

Step 2:- Deriving the unconstrained form

Now, we can transform the constrained problem into an unconstrained problem by minimizing the lagrangian.

By re-arranging lagrangian, we get

$$\mathcal{L}(w, \lambda) = \sum_{n=1}^{N} (f(x_n; w) - t_n)^2 + \lambda \|w\|_p^p - \lambda\gamma$$

since γ is a constant it doesn't affect optimization over w. we can drop $-\lambda\gamma$

$$\min_w \sum_{n=1}^{N} (f(x_n; w) - t_n)^2 + \lambda \|w\|_p^p$$

Thus, we get the transformed constrained problem into the following unconstrained problem:

$$\min_w \sum_{n=1}^{N} (f(x_n; w) - t_n)^2 + \lambda \|w\|_p^p$$

The above is the form of regularized least squares problem, where λ controls the strength of the regularization and $\|w\|_p^p$ is the regularization term

2) Relationship between Hyperparameters λ and γ :-

The hyperparameter λ in the constrained formulation controls the strength of the regularization, while γ is a threshold on the norm of the parameter vector w in the constrained formulation.

* λ and γ are inversely related specifically as λ increases the regularization term $\lambda \|w\|_p^p$ becomes more dominant, forcing the norm of w to decrease. Hence, a large λ implies a small γ i.e., a smaller upper bound on the l_p-norm of w.

* Conversely, if γ is large (the constraint is loose) this implies that the norm of w can be larger, so λ should be small to allow for less penalization.

Problem 5:-

we are proving the convergence result for gradient descent on convex functions with lipschitz continuous gradients. The problem asks us to show that after k iterations of gradient descent with a fixed learning rate $\alpha \leq \frac{1}{L}$, the function satisfies the inequality:

$$f(x^{(k)}) - f(x^*) \leq \frac{1}{2\alpha k} \|x^{(0)} - x^*\|_2^2$$

where $f(x^*)$ is the optimal value of the convex function f.

Step 1:- Prove $f(x^{(k+1)}) \leq f(x^{(k)})$

The iterative update rule for gradient descent is:

$$x^{(k+1)} = x^{(k)} - \alpha \nabla f(x^{(k)})$$

This says that we move in the direction of negative gradient scaled by a step size α.

we use the Lipschitz continuity of the gradient to derive a bound for $f(x^{(k+1)})$.

From the lipschitz property:-

$$f(x^{(k+1)}) \leq f(x^{(k)}) + \nabla f(x^{(k)})^T (x^{(k+1)} - x^{(k)}) + \frac{L}{2} \|x^{(k+1)} - x^{(k)}\|_2^2$$

Substitute $x^{(k+1)} - x^{(k)} = -\alpha \nabla f(x^{(k)})$ into inequality:

$$f(x^{(k+1)}) \leq f(x^{(k)}) - \alpha \|\nabla f(x^{(k)})\|_2^2 + \frac{L\alpha^2}{2} \|\nabla f(x^{(k)})\|_2^2$$

This simplifies to:

$$f(x^{(k+1)}) \leq f(x^{(k)}) - \left(\alpha - \frac{L\alpha^2}{2}\right) \|\nabla f(x^{(k)})\|_2^2$$

For $\alpha \leq \frac{1}{L}$, we have $\alpha - \frac{L\alpha^2}{2} \geq \frac{\alpha}{2}$.

So,

$$f(x^{(k+1)}) \leq f(x^{(k)}) - \frac{\alpha}{2} \|\nabla f(x^{(k)})\|_2^2$$

This shows that the function value decreases after each iteration

Step 2:- Prove $f(x^{(k+1)}) - f(x^*) \leq$

$$\frac{1}{2\alpha} \left(\|x^{(k)} - x^*\|_2^2 - \|x^{(k+1)} - x^*\|_2^2\right)$$

We want to bound the difference $f(x^{(k)}) - f(x^*)$, where x^* is the optimal solution. From the convexity of f, we have:

$$f(x^{(k)}) - f(x^*) \leq \nabla f(x^{(k)})^T (x^{(k)} - x^*)$$

Now use the gradient descent update $x^{(k+1)} = x^{(k)} - \alpha \nabla f(x^{(k)})$. The difference between $x^{(k+1)}$ and x^* is:

$$x^{(k+1)} - x^* = x^{(k)} - \alpha \nabla f(x^{(k)}) - x^*$$

Taking the squared norm of both sides

$$\|x^{(k+1)} - x^*\|_2^2 = \|x^{(k)} - x^*\|_2^2 - 2\alpha \nabla f(x^{(k)})^T (x^{(k)} - x^*) + \alpha^2 \|\nabla f(x^{(k)})\|_2^2$$

Using the convexity bound $\nabla f(x^{(k)})^T (x^{(k)} - x^*) \geq f(x^{(k)}) - f(x^*)$, we get:

$$\|x^{(k+1)} - x^*\|_2^2 \leq \|x^{(k)} - x^*\|_2^2 - 2\alpha (f(x^{(k)}) - f(x)) + \alpha^2 \|\nabla f(x^{(k)})\|_2^2$$

Rearranging the terms:-

$$2\alpha (f(x^{(k)}) - f(x^*)) \leq \|x^{(k)} - x^*\|_2^2 - \|x^{(k+1)} - x^*\|_2^2$$

Dividing by 2α, we obtain:

$$f(x^{(k)}) - f(x^*) \leq \frac{1}{2\alpha} (\|x^{(k)} - x^*\|_2^2 - \|x^{(k+1)} - x^*\|_2^2)$$

Step 3:- Prove $\sum_{k=1}^{K} (f(x^{(k)}) - f(x^*)) \leq \frac{1}{2\alpha} \|x^{(0)} - x^*\|_2^2$

Now, sum the inequality from step 2 over all iterations from $k = 1$ to K:

$$\sum_{k=1}^{K} (f(x^{(k)}) - f(x^*)) \leq \frac{1}{2\alpha} (\|x^{(0)} - x^*\|_2^2 - \|x^{(k)} - x^*\|_2^2)$$

since $\|x^{(k)} - x^*\|_2^2 \geq 0$, we conclude

$$\sum_{k=1}^{K} (f(x^{(k)}) - f(x^*)) \leq \frac{1}{2\alpha} \|x^{(0)} - x^*\|_2^2$$

Dividing both sides by K, we get the average decrease in the function value:

$$\frac{1}{K} \sum_{k=1}^{K} (f(x^{(k)}) - f(x^*)) \leq \frac{1}{2\alpha K} \|x^{(0)} - x^*\|_2^2$$

since $f(x^{(K)}) \leq \frac{1}{K} \sum_{k=1}^{K} f(x^{(k)})$, this implies:

$$f(x^{(K)}) - f(x^*) \leq \frac{1}{2\alpha K} \|x^{(0)} - x^*\|_2^2$$

we have proven that gradient descent with a step size $\alpha \leq \frac{1}{L}$ converges to the optimal solution at a rate of $O(1/K)$, with a final bound:

$$f(x^{(K)}) - f(x^*) \leq \frac{1}{2\alpha K} \|x^{(0)} - x^*\|_2^2$$