Homework - October 1, 2024

Student: Ayrton Chilibeck, achilibe@ualberta.ca
Lecturer: Lili Mou, UoA.F24.466566@gmail.com

> **Problem 1: Convexity across a Function Composition**
>
> Let f and g be convex functions on the same domain. Prove that $f+g$ is also a convex function.

Recall the definition of concavity:

$$f(y) \geq f(x) + \nabla f(x)^T (y-x)$$

If both f and g follow this definition, then we can write

$$f(y) + g(y) \geq f(x) + \nabla f(x)^T (y-x) + g(x) + \nabla g(x)^T (y-x)$$

I define $h = f + g$, so that we can write

$$h(y) \geq h(x) + \nabla f(x)^T (y-x) + \nabla g(x)^T (y-x)$$

Now, we can recall the definition of the gradient ∇:

$$\nabla f = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}$$

since we are multiplying a single term by this collumn matrix, I write

$$h(y) \geq h(x) + (\nabla f(x)^T + \nabla g(x)^T)(y-x)$$

and we can further simplify to

$$h(y) \geq h(x) + (\nabla h(x)^T)(y-x)$$

as required. We see that the sum of two convex functions will result in another convex function. □

> **Problem 2: Closed Form L2-Penalized MSE**
>
> Give the closed-form solution for an l2-penalized mean square error.

Recall the definition of the L2 penalized MSE:

$$J(\vec{w}^{(t)}) = \frac{1}{2M} \sum_{i=1}^{M} \left(sum_{i=0}^{d} w_i x_i^{(m)} - t^{(m)} \right)^2 + \lambda \sum_{i=0}^{d} w_i^2$$

we can rewrite this in matrix notation as follows:

$$J(\vec{w}^{(t)}) = \frac{1}{2M} (X^\top w - \vec{t})^\top (X^\top w - \vec{t}) + \lambda |\vec{w}_i|_2^2$$

Our goal is to compute the optimal $\vec{w}$ for our algorithm, which is one such that we minimize the loss. If we take the gradient with respect to $\vec{w}$, we can solve for the optimal $\vec{w}$ as follows:

$$\begin{aligned}
\nabla_w J(\vec{w}^{(t)}) &= \frac{1}{M} X^\top (X\vec{w} - \vec{t}) + 2\lambda\vec{w} \\
\vec{0} &= X^\top (X\vec{w} - \vec{t}) + M\lambda\vec{w} && \text{F.O. Condition} \\
\vec{w} &= \left(X^\top X + M\lambda I\right)^{-1} (X^\top \vec{t})
\end{aligned}$$

Which gives us the optimal w for our algorithm. □

Problem 3: Gradient Optimization for L1-Penalized MSE

Give a gradient-based optimization algorithm for an l1-penalized mean square error:

Recall the general form for a gradient descent optimization algorithm:

1. Initialize the weights
2. Check the gradient at the starting point
3. modify the weights according to a learning rate
4. repeat starting at step 2 until the loss is below a certain threshold

We need to know how to calculate the gradient of the L1 loss in order to follow the steps above, so we can compute it as follows:

$$\begin{aligned}
L(\vec{w}) &= \frac{1}{2M} \sum_{i=1}^{n} \left(\sum_{i=0}^{d} x_i w_i^{(m)} - t^{(m)} \right) + \lambda \sum_{i=0}^{d} |w_i| \\
L(\vec{w}) &= \frac{1}{2M} \left(X\vec{w} - \vec{t}\right)^\top \left(X\vec{w} - \vec{t}\right) + \lambda|\vec{w}| \\
\nabla_w L(\vec{w}) &= \frac{1}{M} X^\top (X\vec{w} - \vec{t})^1
\end{aligned}$$

We also need to recall the proximal operator:

$$\text{prox}(w, \tau) = \begin{cases} w - \tau & \text{if } w > \tau \\ 0 & \text{if } |w| \leq \tau \\ w + \tau & \text{if } w < -\tau \end{cases}$$

With these results, we can provide the pseudocode for the gradient-based optimization of $\vec{w}$ using the L1 penalized MSE as shown in algorithm 1

```
Input: Initial weights $\vec{w}^{(0)}$, learning rate $\eta$, maximum iterations $T$
for $t = 0$ to $T - 1$ do
    $\nabla_w L(\vec{w}) \leftarrow \frac{1}{M} X^\top (X\vec{w} - \vec{t})$;
    $\vec{w}^{(t+1)} \leftarrow \vec{w}^{(t)} - \eta \nabla J(\vec{w}^{(t)})$;
    $\vec{w}^{(t+1)} \leftarrow \text{prox}(\vec{w}^{(t+1)}, \lambda\eta)$;
end
```

Algorithm 1: Gradient-Based Optimization Algorithm

Problem 4: Probabilistic Interpretation of L1-Penalized MSE

Give a probabilistic interpretation for the L1-penalized MSE loss.

Recall the definition of linear regression in a probabilistic context:

$$y_i = X_i^\top \vec{w} + \epsilon_i$$

Where $\epsilon \ \mathcal{N}(0, \sigma^2)$. This means that $X^\top w$ should provide the expected value of $f(y)$ with some error given by a normal distribution. Since ϵ follows the Gaussian normal distribution, we can write

$$P(y|X, w) = \Pi_{i=0}^{d} \mathcal{N}(X_i^t \vec{w}, \sigma^2)$$

We then take the logarithm of this to reduce the problem to a summation

$$\begin{aligned}
\log\left(P(y|X, w)\right) &= \sum_{i=0}^{d} \left[\log\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) - \frac{X_i^\top \vec{w} - t_i}{2\sigma^2} \right] \\
&= \frac{-}{d} 2 \log\left(2\pi\sigma^2\right) - \sum_{i=0}^{d} \left[\frac{X_i^\top \vec{w} - t_i}{2\sigma^2} \right] \\
&= \frac{-}{d} 2 \log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2} \sum_{i=0}^{d} \left(X_i^\top \vec{w} - t_i \right)^2
\end{aligned}$$

In order to apply the L1 regularization to this problem, we can add a regularizing factor to the expression:

$$P(\vec{w}) \alpha \exp\left(-\lambda ||\vec{w}||_1\right)$$

We can find $P(\vec{w}|X, \vec{y})$ using Bayes' theorem

$$P(\vec{w}|X, \vec{t}) \alpha P(\vec{t}|X, \vec{w}) \cdot P(\vec{w})$$

and since we know the distribution of $P(w)$, we can follow the steps to find the logarithmic version of $P(\vec{w}|X, \vec{t})$, appending the multiplication of $P(w)$ to get

$$P(\vec{w}|X, \vec{t}) = -\frac{d}{2} \log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2} \sum_{i=0}^{d} \left(X_i^\top \vec{w} - t_i \right) - \lambda ||\vec{w}||_1$$

If we minimize this function, then we get the same equation as the L1 loss:

$$\begin{aligned}\hat{w} = \text{argmin}_w P(\vec{w}|X,\vec{t}) &= \text{argmin}_w \left[-\frac{d}{2}\log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2}\sum_{i=0}^{d}\left(X_i^\top \vec{w} - t_i\right)^2 - \lambda||\vec{w}||_1\right] \\ &= \text{argmin}_w \left[\frac{1}{2\sigma^2}\sum_{i=0}^{d}\left(X_i^\top \vec{w} - t_i\right)^2 - \lambda||\vec{w}||_1\right]\end{aligned}$$

Conceptually, this means that minimizing the loss function and maximizing the expectation function of the probability distribution are equivalent, so minimizing the loss function is an accurate way to obtain an optimal weight matrix w for a model. □