Machine Learning Assignment 2
Calculations

Alireza Dastmalchi Saei

Student ID: *993613026*

University: *University of Isfahan*

Course: *Machine Learning Course*

February 2, 2024

1 Question 1

Bayes' theorem is given by:

$$p(w|t_1,\ldots,t_n,x_1,\ldots,x_N) = \frac{p(t_1,\ldots,t_N|x_1,\ldots,x_N,w)\cdot p(w)}{p(t_1,\ldots,t_N,x_1,\ldots,x_N)}$$

The likelihood function, assuming i.i.d data, is given by:

$$\begin{aligned} likelihood &= p(t_1,\ldots,t_N|x_1,\ldots,x_N,w,b) \\ &= \prod_{i=1}^{N} p(t_i|x_i,w,b) \quad \text{(i.i.d)} \\ &= \prod_{i=1}^{N} \left[\sigma(w^T x_i+b)^{t_i}\cdot(1-\sigma(w^T x_i+b))^{1-t_i}\right] \end{aligned}$$

Minimizing the negated likelihood is equivalent to maximizing the likelihood function. By minimizing the negated likelihood, we can find the optimal values of the parameters (w) and (b) that maximize the probability of observing the given targets $(t_1,\ldots,t_N)$ based on the input features $(x_1,\ldots,x_N)$. Minimizing the negated log-likelihood is equivalent to maximizing the likelihood function. The log-likelihood function is a smooth, continuous, and differentiable function, making it just the thing we need for optimizations that require differentiability:

$$\begin{aligned} loss &= -\ln(p(\mathbf{w}|t_1,\ldots,t_n,x_1,\ldots,x_N)) \\ &= -\ln(p(t_1,\ldots,t_N|x_1,\ldots,x_N,\mathbf{w})*p(\mathbf{w})) \\ &= -(\ln(p(t_1,\ldots,t_N|x_1,\ldots,x_N,\mathbf{w}))+\ln(p(\mathbf{w}))) \\ &= -\ln(p(t_1,\ldots,t_N|x_1,\ldots,x_N,\mathbf{w}))-\ln(p(\mathbf{w})) \end{aligned}$$

Note: Based on Bayes' Theorem mentioned above, we can ignore denominator $p(t_1,\ldots,t_n,x_1,\ldots,x_N)$ due to being a constant and our aim is maximizing weights and biases. As given in question, we count $p(w)$ as the regularization term.

Part1: Calculation and simplification of likelihood: Assuming the following

$$\begin{cases} z = \mathbf{w}^T x_i + b \\ p(t=1|x_i, \mathbf{w}, b) = \sigma(z) \end{cases}$$

We have:

$$\begin{aligned}
\ln(p(t_1, \ldots, t_N | x_1, \ldots, x_N, \mathbf{w})) &= ln(\prod_i^N p(t_i|\mathbf{w})) \\
&= \sum_i^N \ln(p(t_i|\mathbf{w})) \quad \text{(i.i.d)} \\
&= \sum_i^N \ln(t_i\sigma(z) + (1-t_i)(1-\sigma(z))) \\
&= \sum_i^N t_i \ln(\frac{1}{1+e^{-z}}) + \sum_i^N (1-t_i)\ln((1-\frac{1}{1+e^{-z}})) \\
&= -\sum_i^M t_i \ln(1+e^{-z}) - \sum_i^N z_i - \sum_i^N \ln(1+e^{-z}) + \sum_i^N t_i z_i + \sum_i^N t_i \ln(1+e^{-z}) \\
&= \sum_i^N (t_i - 1)z + \sum_i^N \ln(1+e^{-z}) \\
&\to \sum_i^N (1-t_i)z - \ln(1+e^{-z}) \quad \text{(negation)}
\end{aligned}$$

Part2: Calculation and simplification of prior:

$$\begin{aligned}
p(\mathbf{w}) &= \mathcal{N}(0, \alpha^{-1}\mathbf{I}) \\
&= \frac{1}{(2\pi)^{(\frac{d}{2})} |\sum|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\mathbf{w}^T(\alpha^{-1}\mathbf{I})^{-1}\mathbf{w}\right)
\end{aligned}$$

We change the name of $\frac{1}{|\sum|^{\frac{1}{2}}}$ to a constant A because it is same for all values. So, we will have:

$$\begin{aligned}
\ln(p(w)) &= -\frac{d}{2}\ln(2\pi) + A - \frac{1}{2}\mathbf{w}^T(\alpha^{-1}\mathbf{I})^{-1}\mathbf{w} \\
-\ln(p(w)) &= \frac{d}{2}\ln(2\pi) - A + \frac{1}{2}\alpha w^T w
\end{aligned}$$

Part3: Combining what we found on part1 and part2, our loss can be defined as:

$$\text{loss} = \sum_{i}^{N}(1 - t_i)z - \sum_{i}^{N}\ln(\sigma(z)) + \frac{d}{2}\ln(2\pi) - A + \frac{1}{2}\alpha w^T w$$

---

b. Derivative with respect to w_i:

$$\frac{\partial Loss}{\partial w_i} = -\sum_{i=1}^{N}\left(t_i - \sigma(w^T x_i + b)\right)x_i + \alpha w_i$$

Derivative with respect to b:

$$\frac{\partial Loss}{\partial b} = -\sum_{i=1}^{N}\left(t_i - \sigma(w^T x_i + b)\right)$$

---

c. Pseudo code for gradient descent:

Algorithm 1 Gradient Descent

```
Initialize weights: w and bias: b
Set learning rate: lr
Set total number of iterations: T
for iteration in range(total_number_of_iterations) do
    Compute the gradient of the loss with respect to weights and bias:
        gradient_w, gradient_b ← compute_gradient(data, labels, w, b)
    Update the weights:
        w ← w − lr × gradient_w
    Update the bias:
        b ← b − lr × gradient_b
end for
```

2 Question 2

$$\begin{cases} \text{Model 1:} & P(y=1|\mathbf{x}, w_1, w_2) = \text{sigmoid}(w_1 x_1 + w_2 x_2) \\ \text{Model 2:} & P(y=1|\mathbf{x}, w_1, w_2) = \text{sigmoid}(w_0 + w_1 x_1 + w_2 x_2) \end{cases}$$

a. In Model1, if the data point is [0, 0], the label will be determined solely by the sigmoid function applied to $w_1.0 + w_2.0$, which simplifies to $P(y = 1|\mathbf{x}, w_1, w_2) = sigmoid(0) = 0.5$, so the likelihood of the Model1 doesn't depend on values of weights. But it does matter in Model2.

---

b. The formula of derivative equals after simplifying:

$$p = \frac{1}{1 + \exp -(w_1 x_1 + w_2 x_2)}$$

$$\frac{\partial L}{\partial w_j} = \sum_i^3 x_j^{(i)} \cdot (y^{(i)} - p)$$

The learning phase with gradient descent:

$$w_{\text{new}} \leftarrow w_{\text{old}} - \text{lr} \cdot \frac{\partial L}{\partial w_j}$$

$$w_{\text{new}} \leftarrow w_{\text{old}} - \text{lr} \cdot \left(\sum_i^3 x_j^{(i)} \cdot \left(y^{(i)} - p \right) \right)$$

If the data point is [0, 0], then $x_j^{(i)}$ for both $j = 0$ and $j = 1$ will be 0, because both features of the data point are zero. Consequently, the entire summation term becomes zero, regardless of the label or the other values in the sum. So, the gradient descent update for the weights will not be affected by this data point, and the weights will remain unchanged after this specific update. (Likelihood of the model1 does not depend on the value of $\mathbf{w}$, but it does matter in model2)

---

c. If we use the loss below (with regularization):

$$\text{Loss} = -\sum_{i=1}^{N} \left[y^{(i)} \log(\text{sigmoid}(\mathbf{w}^T \mathbf{x}^{(i)})) + (1 - y^{(i)}) \log(1 - \text{sigmoid}(\mathbf{w}^T \mathbf{x}^{(i)})) \right] + \frac{\lambda}{2} \|\mathbf{w}\|^2$$

Weights (w_1, w_2) will be updated as following:

$$\frac{\partial \text{Loss}}{\partial w_1} = -\sum_{i=1}^{3} \Big[(y^{(i)} - \text{sigmoid}(\mathbf{w}^T \mathbf{x}^{(i)})) x_1^{(i)} \Big] + \lambda w_1$$

$$\frac{\partial \text{Loss}}{\partial w_2} = -\sum_{i=1}^{3} \Big[(y^{(i)} - \text{sigmoid}(\mathbf{w}^T \mathbf{x}^{(i)})) x_2^{(i)} \Big] + \lambda w_2$$

---

d. If we use the binary cross-entropy loss (with regularization):

$$\text{Loss} = -\sum_{i=1}^{N} \Big[y^{(i)} \log(\text{sigmoid}(\mathbf{w}^T \mathbf{x}^{(i)})) + (1 - y^{(i)}) \log(1 - \text{sigmoid}(\mathbf{w}^T \mathbf{x}^{(i)})) \Big] + \frac{\lambda}{2} \|\mathbf{w}\|^2$$

Weights (w_0, w_1, w_2) will be updated as following:

$$\frac{\partial \text{Loss}}{\partial w_0} = -\sum_{i=1}^{3} \Big[(y^{(i)} - \text{sigmoid}(\mathbf{w}^T \mathbf{x}^{(i)})) \Big] + \lambda$$

$$\frac{\partial \text{Loss}}{\partial w_1} = -\sum_{i=1}^{3} \Big[(y^{(i)} - \text{sigmoid}(\mathbf{w}^T \mathbf{x}^{(i)})) x_1^{(i)} \Big] + \lambda w_1$$

$$\frac{\partial \text{Loss}}{\partial w_2} = -\sum_{i=1}^{3} \Big[(y^{(i)} - \text{sigmoid}(\mathbf{w}^T \mathbf{x}^{(i)})) x_2^{(i)} \Big] + \lambda w_2$$

---

e.

$$\log l(\mathbf{w}) \approx \sum_i \frac{1}{2} y^{(i)} \mathbf{w}^T x^{(i)} - \frac{\lambda}{2} ||w||^2$$

$$\frac{\partial logl(\mathbf{w})}{\partial w_1} \approx \frac{1}{2} \sum_i y^{(i)} x_1^{(i)} - \lambda w_1 = 0$$

$$\frac{\partial logl(\mathbf{w})}{\partial w_2} \approx \frac{1}{2} \sum_i y^{(i)} x_2^{(i)} - \lambda w_2 = 0$$

$$\mathbf{w} = \frac{1}{2\lambda} \sum_i y^{(i)} \mathbf{x}^{(i)}$$

By increasing λ the regularization terms will get stronger (Penalty for bigger weights will soar). Accordingly, weight values will decrease. Increasing λ to a reasonable amount, will help reduce the risk of over-fitting.

3 Question 3

Linear Regression Model:

$$y = ax^2 + bx + c + \epsilon \sim \mathcal{N}(0, \sigma^2)$$

Probability Distribution:

$$P(y|x, a, b, c) = \mathcal{N}(ax^2 + bx + c, \sigma^2)$$

Likelihood Function:

$$L(a, b, c, \sigma^2) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - (ax_i^2 + bx_i + c))^2}{2\sigma^2}\right)$$

If we take logarithm from both sides, our log likelihood will be:

$$\log(L(a, b, c, \sigma^2)) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - (ax_i^2 + bx_i + c))^2$$

To estimate the parameters a, b, and c for the quadratic regression model, we take the partial derivatives of the log-likelihood with respect to each parameter and set them to zero.

$$\begin{aligned}
\frac{\partial \log(L)}{\partial a} &= \frac{1}{\sigma^2}\sum_{i=1}^{n} x_i^2(y_i - (ax_i^2 + bx_i + c)) = 0 \\
&= \frac{1}{\sigma^2}\sum_{i=1}^{n} x_i^2 y_i - \frac{1}{\sigma^2}\sum_{i=1}^{n}(ax_i^4 + bx_i^3 + cx_i^2) \\
&= \frac{1}{\sigma^2}\sum_{i=1}^{n} x_i^2 y_i - \frac{1}{\sigma^2}\sum_{i=1}^{n} ax_i^4 - \frac{1}{\sigma^2}\sum_{i=1}^{n} bx_i^3 - \frac{1}{\sigma^2}\sum_{i=1}^{n} cx_i^2 \\
&= \frac{\sum_{i=1}^{n} x_i^2 y_i}{\sigma^2} - a\frac{\sum_{i=1}^{n} x_i^4}{\sigma^2} - b\frac{\sum_{i=1}^{n} x_i^3}{\sigma^2} - c\frac{\sum_{i=1}^{n} x_i^2}{\sigma^2} = 0
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \log(L)}{\partial b} &= \frac{1}{\sigma^2}\sum_{i=1}^{n} x_i(y_i - (ax_i^2 + bx_i + c)) = 0 \\
&= \frac{\sum_{i=1}^{n} x_i y_i}{\sigma^2} - a\frac{\sum_{i=1}^{n} x_i^3}{\sigma^2} - b\frac{\sum_{i=1}^{n} x_i^2}{\sigma^2} - c\frac{\sum_{i=1}^{n} x_i}{\sigma^2} = 0
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \log(L)}{\partial c} &= \frac{1}{\sigma^2}\sum_{i=1}^{n}(y_i - (ax_i^2 + bx_i + c)) = 0 \\
&= \frac{\sum_{i=1}^{n}(y_i - ax_i^2 - bx_i)}{\sigma^2} - nc = 0
\end{aligned}$$

An estimation to value of b will be:

$$b = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i^2}$$

An estimation to value of a will be:

$$a = \frac{\sum_{i=1}^{n} x_i^2 y_i}{\sum_{i=1}^{n} x_i^4}$$

An estimation to value of c will be:

$$c = \frac{1}{n} \sum_{i=1}^{n} (y_i - a x_i^2 - b x_i)$$

Now for parameter σ we have:

$$\begin{aligned} \frac{\partial \log(L)}{\partial \sigma^2} &= -\frac{n}{2\sigma^2} + \frac{1}{2(\sigma^2)^2} \sum_{i=1}^{n} (y_i - (a x_i^2 + b x_i + c))^2 = 0 \\ &= -\frac{n}{2\sigma^2} + \frac{1}{2(\sigma^2)^2} \sum_{i=1}^{n} (y_i - a x_i^2 - b x_i - c)^2 = 0 \end{aligned}$$

The σ will be:

$$\begin{aligned} \frac{n}{2\sigma^2} &= \frac{1}{2(\sigma^2)^2} \sum_{i=1}^{n} (y_i - a x_i^2 - b x_i - c)^2 \\ \sigma^2 &= \frac{\sum_{i=1}^{n} (y_i - a x_i^2 - b x_i - c)^2}{n} \end{aligned}$$

b. With training data:

$$D = (5, 8), (1, 4), (3, -2), (0, -8)$$

We can calculate model parameters using Maximum Likelihood Estimation (MLE):

$$a = \frac{(5^2 \cdot 8 + 1^2 \cdot 4 + 3^2 \cdot (-2) + 0^2 \cdot (-8))}{(5^4 + 1^4 + 3^4 + 0^4)} \approx 0.263$$

$$b = \frac{(5 \cdot 8 + 1 \cdot 4 + 3 \cdot (-2) + 0 \cdot (-8))}{(5^2 + 1^2 + 3^2 + 0^2)} \approx 1.085$$

$$c = \frac{(8 + 4 - 2 - 8) - a(5^2 + 1^2 + 3^2 + 0^2) - b(5 + 1 + 3 + 0)}{4} \approx -4.242$$

$$\sigma^2 = \frac{(8 - a \cdot 5^2 - b \cdot 5 - c)^2 + (4 - a \cdot 1^2 - b \cdot 1 - c)^2}{4}$$
$$+ \frac{((-2) - a \cdot 3^2 - b \cdot 3 - c)^2 + ((-8) - a \cdot 0^2 - b \cdot 0 - c)^2}{4} \approx 15.427$$

4 Question 4

As given likelihood and prior are like below:

$$\text{Likelihood} = P(y|\mathbf{x}, \mathbf{w}) = \mathcal{N}(\mathbf{xw}, \sigma^2 I) = \prod_{i=1}^{|D|} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(y_i - \mathbf{w^T x_i})^2}{2\sigma^2}\right)$$

$$\text{Prior} = P(\mathbf{w}) = \mathcal{N}(0, \sigma_0^2 I) = \prod_{j=1}^{|w|} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(\mathbf{W}_j)^2}{2\sigma_0^2}\right)$$

For finding MAP for weights we can use Bayes Rule:

$$P(w|D) = \frac{P(D|w) \cdot P(w)}{P(D)}$$

We can take logarithms from both sides:

$$\begin{aligned}
\log(P(\mathbf{w}|D)) &= \log(P(D|\mathbf{w})) + \log(P(\mathbf{w})) - \log(P(D)) \\
&= \text{argmax}_w P(\mathbf{w}|D) \\
&= \text{argmax}_w \log(P(\mathbf{w}|D)) \\
&= \text{argmax}_w \log(P(D|\mathbf{w}) + \log(p(\mathbf{w})))
\end{aligned}$$

Now by replacing given likelihood and prior for the expression above:

$$\begin{aligned}
\log(\text{Likelihood}) &= \log(P(y|\mathbf{x}, \mathbf{w})) \\
&= \log(\mathcal{N}(\mathbf{xw}, \sigma^2 I)) \\
&= \sum_{i=1}^{|D|} \left(\log\left(\frac{1}{\sigma\sqrt{2\pi}}\right) - \frac{(y_i - \mathbf{w^T x_i})^2}{2\sigma^2}\right) \\
&= \sum_{i=1}^{|D|} \log\left(\frac{1}{\sigma\sqrt{2\pi}}\right) - \sum_{i=1}^{|D|} \frac{(y_i - \mathbf{w^T x_i})^2}{2\sigma^2} \\
&= |D| \log\left(\frac{1}{\sigma\sqrt{2\pi}}\right) - \frac{1}{2\sigma^2} \sum_{i=1}^{|D|} (y_i - \mathbf{w^T x_i})^2.
\end{aligned}$$

$$\begin{aligned}
\log(\text{Prior}) &= \sum_{j=1}^{|w|} \left(\log\left(\frac{1}{\sigma_0\sqrt{2\pi}}\right) - \frac{(\mathbf{W}_j)^2}{2\sigma_0^2}\right) \\
&= |w| \log\left(\frac{1}{\sigma_0\sqrt{2\pi}}\right) - \frac{1}{2\sigma_0^2} \sum_{j=1}^{|w|} (\mathbf{W}_j)^2.
\end{aligned}$$

So our new expression for MAP will be like below:

$$\begin{aligned}\text{MAP}(\mathbf{w}|D) &= \log(\text{Likelihood}) + \log(\text{Prior}) \\ &= \left(|D|\log\left(\frac{1}{\sigma\sqrt{2\pi}}\right) - \frac{1}{2\sigma^2}\sum_{i=1}^{|D|}(y_i - \mathbf{w^T x_i})^2\right) + \left(|w|\log\left(\frac{1}{\sigma_0\sqrt{2\pi}}\right) - \frac{1}{2\sigma_0^2}\sum_{j=1}^{|w|}(\mathbf{W}_j)^2\right) \\ &= |D|\log\left(\frac{1}{\sigma\sqrt{2\pi}}\right) - \frac{1}{2\sigma^2}\sum_{i=1}^{|D|}(y_i - \mathbf{w^T x_i})^2 + |w|\log\left(\frac{1}{\sigma_0\sqrt{2\pi}}\right) - \frac{1}{2\sigma_0^2}\sum_{j=1}^{|w|}(\mathbf{W}_j)^2.\end{aligned}$$

To find the maximum value of the MAP (Maximum A Posteriori) estimate, we take the derivative of the expression with respect to $\mathbf{w}$ and set it to zero. The derivative is taken to find the critical points where the function's slope is zero. Let's denote the MAP as $F(\mathbf{w})$ for simplicity. The process can be mathematically expressed as follows:

$$\begin{aligned}F(\mathbf{w}) &= \log(\text{Likelihood}) + \log(\text{Prior}) \\ &= |D|\log\left(\frac{1}{\sigma\sqrt{2\pi}}\right) - \frac{1}{2\sigma^2}\sum_{i=1}^{|D|}(y_i - \mathbf{w^T x_i})^2 + |w|\log\left(\frac{1}{\sigma_0\sqrt{2\pi}}\right) - \frac{1}{2\sigma_0^2}\sum_{j=1}^{|w|}(\mathbf{W}_j)^2\end{aligned}$$

$$\begin{aligned}\frac{\partial F(\mathbf{w})}{\partial \mathbf{w}} &= -\frac{1}{\sigma^2}\sum_{i=1}^{|D|}(y_i - \mathbf{w^T x_i})\mathbf{x_i} - \frac{1}{\sigma_0^2}\sum_{j=1}^{|\mathbf{w}|}\mathbf{w_j} \\ &\Rightarrow -\frac{1}{\sigma^2}\sum_{i=1}^{|D|}(y_i - \mathbf{w^T x_i})\mathbf{x_i} - \frac{1}{\sigma_0^2}\sum_{j=1}^{|\mathbf{w}|}\mathbf{w_j} = 0 \\ &\Rightarrow \sigma_0^2\sum_{i=1}^{|D|}(y_i - \mathbf{w^T x_i})\mathbf{x_i} + \sigma^2\mathbf{w} = 0 \\ &\Rightarrow \sigma_0^2\sum_{i=1}^{|D|}(y_i - \mathbf{w^T x_i})\mathbf{x_i} = -\sigma^2\mathbf{w} \\ &\Rightarrow \sum_{i=1}^{|D|}(y_i - \mathbf{w^T x_i})\mathbf{x_i} = -\frac{\sigma^2}{\sigma_0^2}\mathbf{w} \\ &\Rightarrow \sum_{i=1}^{|D|}\mathbf{x_i y_i} - \sum_{i=1}^{|D|}\mathbf{w^T x_i x_i} = -\frac{\sigma^2}{\sigma_0^2}\mathbf{w}\end{aligned}$$

NOTE: As the precision parameter σ in the prior distribution approaches infinity, the maximum a posteriori (MAP) estimation converges toward the maximum likelihood estimation (MLE), indicating a diminishing influence of the increasingly uninformative prior.

5 Question 5

a. **True** $\rightarrow$ Applying the properties of logarithms, we can simplify the expression and we can transform the given model into a linear form, which would allow us to apply linear regression to estimate the maximum likelihood of parameter (a).

The equation will be simplified as follows:

$$\begin{aligned} y_i &= \log\left(x_1^{a_1} \cdot e^{a_2}\right) + \epsilon_i \\ &= \log(x_1^{a_1}) + \log(e^{a_2}) + \epsilon_i \\ &= a_1 \log(x_1) + a_2 + \epsilon_i \end{aligned}$$

Now, we have transformed the equation into a linear form in terms of a_1 and a_2, where the dependent variable (y_i) is linearly related to the independent variable $(\log(x_1))$ with parameters $((a_1))$ and (a_2). This allows us to apply linear regression techniques to estimate the maximum likelihood of parameter (a).

---

b. **False** $\rightarrow$ y is not linear in a_1 and a_2 and we cannot transform the equation into linear form using simple transformations.

$$y_i = \log(x_1^{a_1} \cdot e^{a_2} + \epsilon_i)$$

No further simplifications can be done to achieve a linear form.

6 Question 6

Given the information, the model with a degree of 2 (Model A) is likely to perform better on the test data. This is because higher-degree polynomial models, such as the one with degree 5 (Model B), may capture noise in the training data and overfit (high variance/low bias), resulting in poorer generalization to test data.

Model A is expected to have a better balance between fitting the training data and generalizing to the test data (Balanced variance and bias).