Problem 1: Refreshers on Optimization and probability fundamentals.

(a) To find the optimal solution θ^* that minimizes $f(\theta)$, we need to find the critical points of $f(\theta)$ by taking the derivative with respect to θ and setting it to zero. Then, we can determine whether the critical point is a minimum by analyzing the second derivative.

First, let's find the first derivative of $f(\theta)$ with respect to θ:

$$f'(\theta) = \Sigma_{i=1}^{n} - 2w_i(x_i - \theta)$$
$$\Sigma_{i=1}^{n} 2w_i(x_i - \theta) = 0$$
$$\Sigma_{i=1}^{n} w_i x_i - w_i\theta = 0$$
$$\theta^* = \frac{\Sigma_{i=1}^{n} w_i x_i}{\Sigma_{i=1}^{n} w_i}$$

Now let's verify that this is a minimum by analyzing the second derivative:

$$f''(\theta) = \frac{d^2}{d\theta^2}\left(\sum_{i=1}^{n} w_i(x_i - \theta)^2\right) = \frac{d}{d\theta}\left(-2\sum_{i=1}^{n} w_i(x_i - \theta)\right) = 2\sum_{i=1}^{n} w_i$$

Since $w_i > 0$, the second derivative $f''(\theta)$ is always positive, which implies that θ^* is indeed a minimum.

Thus, the optimal solution θ^* that minimizes $f(\theta)$ is:

$$\theta^* = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}$$

Now, if some of the w_i are negative, the function $f(\theta)$ would no longer be convex, as the second derivative $f''(\theta)$ may not be positive for all values of θ. In this case, the function could have multiple local minima, maxima, or saddle points, and the optimization problem becomes more challenging to solve analytically.

(b) To solve this problem, we will consider two separate cases: when Alice is on the committee and when Bob is on the committee. Then we will sum up the number of possibilities from both cases.

Case 1: Alice is on the committee. In this case, we need to choose 3 more members from the remaining 6 researchers (excluding Bob). The number of combinations for this case can be calculated using the combination formula:

$$\binom{6}{3} = \frac{6!}{3!(6-3)!} = \frac{6!}{3!3!} = \frac{6 \times 5 \times 4}{3 \times 2 \times 1} = 20$$

Case 2: Bob is on the committee. Similar to the first case, we need to choose 3 more members from the remaining 6 researchers (excluding Alice). The number of combinations for this case is also:

$$\binom{6}{3} = \frac{6!}{3!(6-3)!} = 20$$

Now, we sum up the possibilities from both cases:

$$\text{Total combinations} = 20 + 20 = 40$$

Thus, there are 40 different ways to form a committee of 4 members from the group of 8 AI researchers, considering the constraint that Alice and Bob cannot be part of the same committee.

(c) (Bayes rule)

- probability of knowing the answer P(K) = p
- probability of not knowing the answer P(NK) = 1-p
- probability of correct answer given that she knows it P(C|K) = 0.99
- probability of correct answer given she does not know & guesses it P(C| NK) = 1/k

Need to find conditional probability P(K|C). From Bayes rule -

$$P(K|C) = \frac{P(C|K) * P(K)}{P(C)}$$
$$P(C) = P(C|K)P(K) + P(C|NK)P(NK)$$
$$\implies P(K|C) = \frac{0.99 * p}{P(C)}$$
$$\implies P(K|C) = \frac{0.99p}{0.99 * p + (1/k) * (1-p)}$$
$$P(K|C) = \frac{0.99p}{0.99 * p + \frac{(1-p)}{k}}$$

(Note: $P(C)$ follows from the law of total probability.)

(d) Need to find gradient for $F(w)$ -

$$F(w) = \Sigma_{i=1}^{n}(x_i^T w - y_i)^2 + \lambda \Sigma_{i=1}^{d} w_i^2 \quad \text{where} \;\; x_1, x_2, x_3... \in \mathbb{R}^d$$

and $y_1, y_2...$ are scalars and λ is a non-negative value. To find gradient for $F(w)$, we need to partial differentiate wrt to each w_i

$$\frac{\partial F(w)}{\partial w_j} = \Sigma_{i=1}^{n} 2(x_i^T w_j - y_i)(x_j^T) + 2\lambda w_j$$
$$\frac{\partial F(w)}{\partial w_j} = \Sigma_{i=1}^{n} 2(x_i^T w_j x_j^T - y_i x_j^T) + 2\lambda w_j$$
$$\frac{\partial f(\mathbf{x})}{\partial w_j} = 2\Sigma_{i=1}^{n}(x_i^T w_j x_j^T - y_i x_j^T) + 2\lambda w_j$$

(e) Calculate gradient for f wrt to vector $\mathbf{x} = (x_1, ..., x_n)^T$ -

$$f(x_1, x_2, x_3...x_n) = log\ \Sigma_{i=1}^{n} \exp(x_i)$$
$$\frac{\partial f(x_1, x_2, x_3...x_n)}{\partial \mathbf{x}} = \frac{\partial (log\ \Sigma_{i=1}^{n} e^{x_i})}{\partial \mathbf{x}}$$
$$\frac{\partial f(\mathbf{x})}{\partial x_j} = \frac{1}{\Sigma_{i=1}^{n} e^{x_i}} * e^{x_j}$$

(partial differentiation w.r.t to each variable x_i)

(f) For any real values of x, let's assume p = $\max_i x_i$ (for all i), then we can say that for any i -

$$e^{x_i} \leq e^p \quad (\text{as } \; e^{x_i} \leq e^{\max_i x_i})$$

(taking summation both sides)

$$\Sigma_{i=1}^{n} e^{x_i} \leq n e^p$$
$$log(\Sigma_{i=1}^{n} e^{x_i}) \leq log(n) + p$$

$$f(x_1, x_2..x_n) \leq log(n) + p \tag{1}$$

Similarly -

$$max_i(e^{x_1}, e^{x_2}..., e^{x_n}) \leq \Sigma_{i=1}^{n} e^{x_i} \tag{2}$$

$$max_i(e^{x_1}, e^{x_2}..., e^{x_n}) \leq log(\Sigma_{i=1}^{n} e^{x_i})$$

Equation (2) is because both LHS and RHS are strictly positive numbers and individual value in a list of numbers will always be lesser than sum of all the list of numbers.

Therefore, combining the above two equations

$$max_i(x_1, x_2...x_n) \leq log\Sigma_{i=1}^{n} e^{x_i} \leq logn + max_i(x_1, x_2...x_n)$$

Problem 2: Coding questions

(a) Code in jupyter notebook.

(b) Worst case complexity: $O(n^2)$ when an A (n*n) matrix multiplied with a dense vector $v \in \mathbf{R}^n$. If we have a sparse matrix A then the time complexity becomes $O(nnz(A))$, i.e number of non-zero elements.

(c) Data-structures used: List, dictionary. Time complexity: O(N).

(d) Time complexity with recursive approach: $O(2^n)$
Time complexity with optimization (with dynamic programming): $O(n)$

References

- Question 1 part (e): https://en.wikipedia.org/wiki/LogSumExp