Anand Natu
06264867

Problem 1: Optimization and Probability

Problem 1a

The sequence is finite, so we differentiate to find the minimum value of θ:

$$\frac{df}{d\theta} = \sum \mathrm{w_i}\,(\theta - x_i) = \sum \mathrm{w_i}\theta - w_i x_i = 0$$

$$\sum \mathrm{w_i}\theta = \sum \mathrm{w_i} x_i$$

$$\boldsymbol{\theta} = \frac{\sum \mathbf{w}_i \boldsymbol{x_i}}{\Sigma \boldsymbol{w_i}} = \sum \mathbf{x_i}$$

If some of the *w* terms are negative in the series, this can create errors with optimization by creating local extrema, which will make the convergent solution of the optimization problem at least partially dependent on initial conditions.

Problem 1b

We want to compare

$$f(x) = \max_{\mathrm{x}} \sum \mathrm{sx_i} \quad vs. \quad g(x) = \sum \max_{\mathrm{x}} sx_i$$

Suppose that there is some x' for which f(x) is maximized. Then, we have

$$f(x) = \sum \mathrm{sx'} \quad vs. \quad g(x) = \sum \max_{\mathrm{x}} sx_i$$

Since the ranges of the summation are equivalent, we then have

$$sx' \quad vs. \quad \max_{\mathrm{X}} sx_i$$

However, because both x and the summation range are constant between f and g, it follows that

$$sx' \le \max_{x} sx_i \;\; for\ all\ x$$

And so therefore

$$\boldsymbol{f(x) \le g(x) \;\; for\ all\ x}$$

Problem 1c

For a six-sided fair dice, the probability of rolling any given number is 1/6. Under the specified conditions, the expected number of points accumulated by the time a 1 is rolled is defined by the following recurrence, which can be solved for P:

$$P = \frac{1}{6}(0) + \frac{3}{6}(P) + \frac{1}{6}(P+b) + \frac{1}{6}(P-a)$$

$$\therefore\ \boldsymbol{P = b - a}$$

Problem 1d

To optimize L(p) at some p=p_{max}, we suppose that dL/dp = 0 for some value p = p_{max}. Taking the logarithm of this function, we see that the optimizing condition becomes

$$L'(p) = \ln\big(L(p)\big)$$

$$\frac{dL'}{dp} = \frac{d}{dp}(\ln\big(L(p)\big) = \frac{1}{L(p)} * \frac{dL}{dp}$$

$$\frac{dL}{dp} = 0\ for\ some\ p = p_{max}$$

$$\therefore \frac{dL'}{dp} = \frac{1}{L(p)} * (0) = 0\ at\ p = p_{max}$$

Therefore, the value of p which maximizes L(p) must also maximize L'(p). Therefore, we take the natural logarithm of both sides. We have

$$L'(p) = \ln(p^4(1-p)^3) = 4\ln(p) + 3\ln(1-p)$$

Differentiating both sides to find the optimizing value of p yields

$$\frac{dL'}{dp} = \frac{4}{p} - \frac{3}{1-p} = 0 \quad for\ some\ p = p_{max}$$

$$3p = 4 - 4p, \therefore \boldsymbol{p = \frac{4}{7}}$$

This value of p represents the optimal "fairness" of the coin to provide the maximum total probability of obtaining the event sequence {H,H,T,H,T,T,H} in a series of 7 tosses. I.e. a coin which has a 4/7 chance of landing heads will maximize the probability of obtaining said sequence.

Problem 1e

Differentiating, we have

$$\nabla f = \Sigma\Sigma\, 2 * \big(a_i w - b_j w\big) * \frac{\delta}{\delta w}\big(a_i w - b_j w\big) + \lambda\Sigma\left(\frac{\delta}{\delta w}\big(w_k^2\big)\right)$$

The L_2-norm term reduces to 0 for all components of the vector w that are not w_k and $2w_k$ otherwise, so

$$\nabla f = \Sigma\Sigma\, 2 * \big(a_i w - b_j w\big) * \frac{\delta}{\delta w}\big(a_i w - b_j w\big) + 2\lambda w$$

Finally, we know that $\frac{\delta}{\delta w}(w) = 1$ for all w_i, so that we have

Anand Natu
06264867

$$\therefore \boldsymbol{\nabla f = \Sigma\Sigma\, 2 * \left(a_i^T w - b_j^T w\right) * \left(a_i^T - b_j^T\right) + 2\lambda w}$$

Problem 2: Complexity

Problem 2a

Let $R(m,n)$ represent the total number of subrectangles of any size that can be drawn within a master rectangle of dimensions $m * n, (m,n) \in I$. For a *1 x n* rectangle, there are *n* rectangles of size *1x1*, *n-1* rectangles of size *1x2,* and so forth. Therefore the total number of rectangles in a *1 x n* rectangle is given by

$$\sum_{i=1}^{n} n = \frac{n(n+1)}{2}$$

And since we have an *n x n* rectangle, the total number of rectangles is given by the square of the above, and since there are 6 rectangles per face, with no constraints on the location or size of the rectangles, we have

$$\left[\left[\left[\frac{n(n+1)}{2}\right]^2\right]^6\right]^2 = \left(\frac{n(n+1)}{2}\right)^{24} \to \boldsymbol{O(n^{24})}$$

Problem 2b

We can only take steps down and right, therefore the cost function is equivalent to the Manhattan distance between the starting point and the destination, such that we have (assuming a starting point of (1,1) and that all movements incur the same unit cost) the following as the most efficient means of computing the cost of movement:

$$c(i,j) = |i-1| + |j-1|$$

The above operation involves two absolute-value operations, two subtractions of one, and one addition of two integers of variable size. Ignoring runtime effects of the size of *i* and *j,* all of these are constant-time operations such that we have O(k) runtime for some constant k.

Problem 2c

Whenever one is at step *n* in a stairwell, the total number of ways they can get to that stair is the sum of the total number of ways they could have gotten to that stair from every preceding stair in the stairwell (since steps can be any number of steps in length), e.g.

$$\#\ of\ step\ combinations\ for\ 3\ steps = S(3) = S(0) + S(1) + S(2) = 1 + 2 + 1 = 4$$

Anand Natu
06264867

Where S(0) above is the number of ways one can *uniquely* reach the top from the bottom of the stairwell (i.e. taking 3 steps), S(1) from the first stair (take one step + one step, take two steps), and S(2) from the second step (take one step).

Therefore, it follows that the total number of ways one can climb a stairwell of length *n* is given by

$$\boldsymbol{S(n) = \sum_{i=1}^{n-1} S(n-i)}$$

This is similar to the Fibonacci relation $F(n) = F(n-1) + F(n-2)$, however in this case we sum the results of all of the preceding sub-problems rather than just the previous 2.

Problem 2d

For vectors $a, b, w \in R^d$

$$f(w) = \sum_{i=1}^{n} \sum_{j=1}^{n} \left(a_i^T w - b_j^T w\right)^2 + \lambda \sum_{k=1}^{d} w_k^2$$

Factoring, we have

$$f(w) = w^2 \left[\sum_{i=1}^{n} \sum_{j=1}^{n} \left(a_i^T - b_j^T\right)^2\right] - \lambda \sum_{k=1}^{d} w_k^2$$

Expanding the quadratic term gives

$$f(w) = w^2 \left[\sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_i^T - a_i b_j^T - b_j a_i^T + b_j b_j^T\right] - \lambda \sum_{k=1}^{d} w_k^2$$

Which in turn gives

$$\boldsymbol{f(w) = w^2 \left[\sum_{i=1}^{n} a_i a_i^T + \sum_{i=1}^{n} a_i \sum_{j=1}^{n} b_j^T + \sum_{i=1}^{n} a_i \sum_{j=1}^{n} b_j^T + \sum_{j=1}^{n} b_j b_j^T\right] - \lambda \sum_{k=1}^{d} w_k^2}$$

In the above, we now have $O(nd^2)$ for the first term, and $O(d^2)$ for the second term in computation of the L_2 norm (whose calculation can be re-used in order to calculate the value of w^2 that appears in the first term).