CSE586: Algorithms Under Uncertainty

Homework 3 Solutions

Ashlesha Gupta (2021380) Divyajeet Singh (2021529)

Solution 1.

Part (a)

We are required to prove that the offline optimal algorithm, $\mathbf{OPT}$, can match all arriving vertices $v_i \in R$ to some unique vertex $u(v_i) \in L$.

Claim 1. $\mathbf{OPT}$ *knows the sequence of vertices* $\langle w_i \rangle$ *that are removed in each step.*

Proof. $\mathbf{OPT}$ has complete knowledge of the input sequence σ of vertices $v_1, v_2, \ldots, v_n$ arriving in the online algorithm. This means that $\mathbf{OPT}$ knows the set of vertices S_i that are adjacent to each vertex v_i in σ. So, given that v_i is connected to all vertices in S_i, $\mathbf{OPT}$ can infer the sequence of vertices $\langle w_i \rangle$ that are removed in each step by using the set difference between S_i and S_{i+1}. □

An optimal algorithm is one that can match the maximum number of vertices in R to vertices in L. To maximize the number of matches, $\mathbf{OPT}$ can match each arriving vertex $v_i \in R$ to the vertex $w_i \in S_{i+1} \setminus S_i$, i.e. we define $u(v_i) \triangleq w_i \; \forall \; v_i \in R$. This ensures that in each step, only unmatched vetices remain in the set S_{i+1}, and hence, each arriving vertex can be matched.

Part (b)

Let W_i be the event that any deterministic online algorithm $\mathcal{A}$ matches the vertex w_i. We must prove that

$$\mathbb{P}[W_i] \leq \frac{1}{n} + \frac{1}{n-1} + \cdots + \frac{1}{n-i+1} = \sum_{j=1}^{i} \frac{1}{n-j+1} \tag{1}$$

To prove this, we simply provide an upper bound of $\frac{1}{n-j+1}$ on the probability of $W_i^{(j)}$, the event that vertex w_i gets matched by any deterministic online algorithm $\mathcal{A}$ in the j^{th} step ($j \leq i$). Certainly, w_i cannot be matched in any step $j > i$ since it is removed from the set S_i after the i^{th} step.
Let $\langle m_i \rangle_{[1:n(\mathcal{A})]}$ to denote the sequence of vertices matched by $\mathcal{A}$.

Claim 2. *At any step j, there are an equal number of sequences $\langle m_k \rangle_{[1:j]}$ such that $m_j = w_l$, i.e. that end in any vertex w_l, where $j \leq l \leq n$.*

Proof. The claim would be trivial if all sequences of vertices were *valid*. However, since after each step, a certain vertex becomes unavailable. We prove that an equal number of *valid* sequences end in any vertex w_l, where $j \leq l \leq n$.

For every sequence $\langle m_k \rangle_{[1:j]}$ that ends in vertex w_j, we can construct a corresponding sequence $\langle m'_k \rangle_{[1:j]}$ that ends in vertex w_l, where $j < l \leq n$, as follows ($\circ$ denotes concatenation)

$$\langle m'_k \rangle_{[1:j]} = \begin{cases} \langle m_k \rangle_{[1:j-1]} \circ w_l & \text{if } w_l \notin \langle m_k \rangle_{[1:j]} \\ \langle m_k \rangle_{[1:p-1]} \circ m_j \circ \langle m_k \rangle_{[p+1:j-1]} \circ m_p & \text{if } w_l = m_p \end{cases} \tag{2}$$

Simply put, replace the last vertex m_j with w_l if w_l has not been matched yet, or swap the last vertex with w_l if w_l was already matched in some round $p \leq j$. Note how the swapping in the second case does not affect the *validity* of the resulting sequence, i.e. the sequence $\langle m'_k \rangle_{[1:j]}$ is still *valid*, since w_l (emphasizing $j < l \leq n$) would still be available for matching in the j^{th} step if $w_l \notin \langle m_k \rangle_{[1:j]}$. Moreover, if the vertex $m_j = w_j$ (by the way we defined construction), was eligible for a match in the j^{th} step, then it would have been eligible for matching in the previous steps as well.
In fact, the above construction is reversible, in the sense that for every sequence that ends in a fixed vertex w_l, where $j \leq l \leq n$, we can construct a corresponding sequence that ends in some other vertex w_k, where $j \leq k \leq n$ ($k \neq l$).
This proves that an equal number of valid seqeunces end in any vertex w_l, where $j \leq l \leq n$. ☐

Corollary 2.1. $\mathbb{P}\left[W_i^{(j)}\right] \leq \frac{1}{n-j+1}$ *for every* $j \leq i$.

Proof. The total probability that $\mathcal{A}$ matches some vertex in round j is clearly at most 1. Since there are at most $n - j + 1$ vertices[1] at which the matching sequence can end (i.e. that can be matched), and claim 2 holds, the probability that $\mathcal{A}$ matches w_i in this round is at most $\frac{1}{n-j+1}$, i.e.

$$\mathbb{P}\left[W_i^{(j)}\right] \leq \frac{1}{n-j+1} \tag{3}$$

Empirically, in a round j, there are at most $n - j + 1$ vertices available for matching. So, if $w_i \notin \langle m_k \rangle_{[1:j]}$, then w_i is available for matching in the j^{th} step. Hence, the probability that it gets picked in this round is at most $\frac{1}{n-j+1}$. ☐

w_i can be matched in any round $j \leq i$. So, the total probability that w_i ends up being matched by $\mathcal{A}$ is

$$\mathbb{P}\left[W_i\right] = \sum_{j=1}^{i} \mathbb{P}\left[W_i^{(j)}\right] \leq \sum_{j=1}^{i} \frac{1}{n-j+1} = \frac{1}{n} + \frac{1}{n-1} + \cdots + \frac{1}{n-i+1} \tag{4}$$

Part (c)

We must prove that there does not exist any online algorithm $\mathcal{A}$ that whose competitive ratio is greater than $\left(1 - \frac{1}{e}\right)$. This can be proved by Yao's principle, i.e. we prove that

$$1 - \frac{1}{e} \leq \frac{\mathbb{E}_{\sigma \in D_I}[\mathbf{\Lambda}^\star(\sigma)]}{\mathbb{E}_{\sigma \in D_I}[\mathbf{OPT}(\sigma)]} \tag{5}$$

Hence, we must find a suitable distribution on the input sequences. The instance will consist of n vertices, where n is a large number. At any round, we send a request to match any vertex v_i which has connections to $u_i, u_{i+1}, \ldots, u_n \in L$, uniformly at random.

[1] We say "at most" $n - j + 1$ vertices available in round j since some of the unmatched vertices may have already been removed from the current set S_j in previous rounds.

Solution 2.

This problem is an extension of the full-feedback n-experts problem covered in class. We are given an expression for the total loss. However, since our algorithm is randomized, we must consider the expected loss. This means that the regret is given by

$$\textsc{Regret}(T) \leq \mathbb{E}\left[\sum_{t=1}^{T} l_{E_t}^{t} + \sum_{t=1}^{T-1} \mathbb{I}_{\{E_t = E_{t+1}\}}\right] - \min_{1\leq i\leq n} \sum_{t=1}^{T} l_i^{(t)} \tag{6}$$

where the expectation is over the randomization of the algorithm, i.e. the experts E_t chosen in each round $t = 1, 2, \ldots, T$.

The proposed randomized algorithm

We propose the randomized algorithm $\mathbf{\Lambda}$ given in Algorithm 1, which is a modification of the Multiplicative Weights Algorithm (MWA). We need to (in expectation) minimize the number of expert switches, and so, the algorithm chooses to stick to the same expert with some probability.

Algorithm 1 Modified MWA algorithm for n-experts with a cost for switching experts

procedure EXPERTS-PROBLEM($\sigma[1:T]$, n, η):
  $w_i^{(1)} \leftarrow 1 \quad 1 \leq i \leq n$
  $W^{(1)} \leftarrow \sum_{i=1}^{n} w_i^{(1)}$
  Sample e_1 from the distribution $\mathcal{P}^{(1)}$ given by $p_i^{(1)} = \frac{w_i^{(1)}}{W^{(1)}} \quad 1 \leq i \leq n$
  for $t \leftarrow 2$ to T **do**
    $w_i^{(t)} \leftarrow w_i^{(t-1)}(1-\eta)^{l_i^{(t)}} \quad 1 \leq i \leq n$
    $W^{(t)} \leftarrow \sum_{i=1}^{n} w_i^{(t)}$
    $e_t \leftarrow \begin{cases} e_{t-1} & \text{with probability } (1-\eta)^{l_{e_{t-1}}^{(t)}} \\ \text{Choose expert } i \text{ with probability distribution } \mathcal{P}^{(t)} & \text{otherwise} \end{cases}$
  end for
end procedure

The intuition behind this algorithm is that it tries to minimize the number of switches by sticking to the same expert e_t with probability $(1-\eta)^{l_{e_t}^{t}}$, which is the weight update for the expert e_t in round t. We expect the best expert to have the lowest loss, and so, we expect the algorithm to stick to the best expert with a higher probability.
This helps the algorithm into phases where the algorithm sticks to the same expert.

Regret Analysis

We prove that $\mathbf{\Lambda}$ has sublinear regret. For this, we make the following claim. By equation (6), we need to bound the expected number of switches in the algorithm. Let the random variable S_t denote the number of switches in the first t rounds. Now, we have

$$\mathbb{E}\left[\sum_{t=1}^{T} l_{E_t}^{t} + \sum_{t=1}^{T-1} \mathbb{I}_{\{E_t = E_{t+1}\}}\right] = \sum_{t=1}^{T} \mathbb{E}\left[l_{E_t}^{t}\right] + \sum_{t=1}^{T-1} \mathbb{E}\left[\mathbb{I}_{\{E_t = E_{t+1}\}}\right] \tag{7}$$

$$= L_{\mathbf{\Lambda}}^{(T)} + \mathbb{E}\left[S_T\right] \tag{8}$$

where $L_{\mathbf{\Lambda}}^{(T)}$ is the expected loss of $\mathbf{\Lambda}$ in T rounds.

Claim 3. *The probability that $\mathbf{\Lambda}$ chooses expert i in round t is equal to the probability with which MWA selects it in the full-feedback n-experts problem, i.e.*

$$\mathbb{P}[e_t = i] = \frac{w^{(i)}}{W^{(t)}} \quad 1 \le i \le n, 1 \le t \le T \tag{9}$$

Proof. In any round $t \ge 2$, $\mathbf{\Lambda}$ chooses expert i either because it was chosen in round $t-1$, or because it was sampled from the distribution $\mathcal{P}^{(t)}$ again when we are not sticking. This gives

$$\mathbb{P}[e_t = i] = \mathbb{P}[e_{t-1} = i] \cdot (1-\eta)^{l_i^{(t)}} + \mathbb{P}[e_{t-1} \ne i] \cdot \frac{w_i^{(t)}}{W^{(t)}} \tag{10}$$

Since this is a recursive claim, we use induction to prove it. Clearly, the claim holds for $t = 1$, since we select e_1 from the distribution $\mathcal{P}^{(1)}$. Assuming the claim holds for $t-1$, we have (following from the previous equation)

$$\begin{aligned}
\mathbb{P}[e_t = i] &= \frac{w_i^{(t-1)}}{W^{(t-1)}} \cdot (1-\eta)^{l_i^{(t)}} + \left(1 - \frac{w_i^{(t-1)}}{W^{(t-1)}} \cdot (1-\eta)^{l_i^{(t)}}\right) \cdot \frac{w_i^{(t)}}{W^{(t)}} && (11)\\
&= \frac{w_i^{(t-1)}}{W^{(t-1)}} \cdot (1-\eta)^{l_i^{(t)}} + \left(1 - \frac{w_i^{(t)}}{W^{(t-1)}}\right) \cdot \frac{w_i^{(t)}}{W^{(t)}} && (12)\\
&= \frac{w_i^{(t)}}{W^{(t-1)}} + \frac{w_i^{(t)}}{W^{(t)}} - \frac{w_i^{(t)}}{W^{(t-1)}} = \frac{w_i^{(t)}}{W^{(t)}} && (13)
\end{aligned}$$

Hence, the claim is proved. □

We now provide a bound on the expected number of switches in the algorithm. There can be only as many switches in the algorithm as many times the algorithm chooses to sample from the distribution $\mathcal{P}^{(t)}$. So, the probability of sampling again in any step $t \ge 2$ is

$$\begin{aligned}
\mathbb{P}[\text{Switching experts at step } t] &= \sum_{j=1}^{n} \mathbb{P}\left[e_{t-1} = j\right] \cdot \mathbb{P}[\text{Switching from expert } j] && (14)\\
&= \sum_{j=1}^{n} \mathbb{P}\left[e_{t-1} = j\right] \cdot \left(1 - (1-\eta)^{l_j^{(t-1)}}\right) && (15)\\
&= 1 - \sum_{j=1}^{n} \frac{w_j}{W^{(t-1)}} \cdot (1-\eta)^{l_j^{(t-1)}} && (16)\\
&= \frac{W^{(t-1)} - W^{(t)}}{W^{(t-1)}} = s_t \quad \text{(say)} && (17)
\end{aligned}$$

From this, we have a bound on the sum of weights at the end of round t.

$$\begin{aligned}
W^{(t)} &= (1 - s_t^t) W^{(t-1)} && (18)\\
\implies W^{(T)} &= W^{(1)} \prod_{t=1}^{T-1} (1 - s_t^t) = n \prod_{t=1}^{T-1} (1 - s_t^t) && (19)\\
&\ge (1-\eta)^{\sum_{i=1}^{T-1} l_i^t} && (20)
\end{aligned}$$

where the last inequality follows from the lecture. Taking natural log on both sides gives

$$\ln n + \sum_{t=1}^{T-1} \ln\left(1 - s_t^t\right) \geq \sum_{t=1}^{T} \ln\left(1 - \eta\right) \tag{21}$$

Claim 3 implies that the expected loss of $\mathbf{\Lambda}$ without counting the cost of switching the experts is the same as that of MWA in the full-feedback n-experts problem. So, we have

$$\begin{aligned}
\textsc{Regret}(T) &\leq L^{(T)}_{\mathbf{MWA}} + \mathbb{E}\left[S_T\right] - \min_{1\leq i\leq n} \sum_{t=1}^{T} l_i^t & (22)\\
&\leq \min_{1\leq i\leq n} \sum_{t=1}^{T} l_i^{(t)} + 2\sqrt{T\ln n} + \frac{W^{(t-1)} - W^{(t)}}{W^{(t-1)}} - \min_{1\leq i\leq n} \sum_{t=1}^{T} l_i^t & (23)\\
&\leq 2\sqrt{T\ln n} + \frac{W^{(t-1)} - W^{(t)}}{W^{(t-1)}} & (24)
\end{aligned}$$

where equation (23) follows from the lecture. So, we finally have (by the above inequalities)

$$\tag{25}$$

Solution 3.

By **Theorem 24.5** in [2], if the loss function $f_t(\mathbf{w})$ is convex, and the regularizer $R(\mathbf{w})$ is σ-strongly convex and follows the Lipschitz conditions with parameter L, then the regret of FTRL is bounded by

$$\textsc{Regret}(T) \leq \max_{\mathbf{u}\in S} R(\mathbf{u}) - R\left(\mathbf{w}^{(1)}\right) + T \cdot \frac{L^2}{\sigma} \tag{26}$$

To provide a bound on the regret of FTRL on the given online linear regression problem with Euclidean regularizer, we solve for the right-hand side of the inequality in (26).
We know (courtsey of lecture) that the Euclidean regularizer is $\frac{1}{\eta}$-strongly convex. So, we already have the value for σ.

$$R(\mathbf{w}) = \frac{1}{2\eta} \sum_{i=1}^{2} w_i^2 = \frac{1}{2\eta} \left\|\mathbf{w}\right\|_2^2 \tag{27}$$

Clearly, the maximum value of $R(\mathbf{w})$ is achieved when $\|\mathbf{w}\|_2 = r$, since $\mathbf{w} \in S$. So, we have

$$\max_{\mathbf{u}\in S} R(\mathbf{u}) = \max_{\mathbf{u}\in S} \frac{1}{2\eta} \left\|\mathbf{u}\right\|_2^2 = \frac{1}{2\eta} r^2 \tag{28}$$

Moreover, $R\left(\mathbf{w}^{(1)}\right) \geq 0$, so we can simply ignore it in the bound (equivalently, we take its minimizer, which is 0). Hence, we have

$$\textsc{Regret}(T) \leq \frac{1}{2\eta} r^2 + T\eta L^2 \tag{29}$$

Now, we find a value for the parameter L for which the loss function $f_t(\mathbf{w})$ satisfies the Lipschitz condition. We use the $\mathcal{L}_2$-norm for the right-hand size of the Lipschitz inequality.

$$\begin{aligned}
\left|f_t(\mathbf{u}) - f_t(\mathbf{v})\right| &\leq L\left\|\mathbf{u} - \mathbf{v}\right\|_2 & (30)\\
\left|\left(u_1 x^{(t)} + u_2 - y^{(t)}\right)^2 - \left(v_1 x^{(t)} + v_2 - y^{(t)}\right)^2\right| &\leq L\sqrt{(u_1 - v_1)^2 + (u_2 - v_2)^2} & (31)\\
&\leq L \cdot 2r \quad \text{since } \max_{\mathbf{u},\mathbf{v}\in S} \|\mathbf{u} - \mathbf{v}\|_2 = 2r & (32)
\end{aligned}$$

We find a value for L by overestimating the left-hand side of the inequality in (30). This gives us a slightly loose bound on L, but it is still valid. For notational convenience, we define $\mathbf{w}^T x^{(t)} \triangleq w_1 x^{(t)} + w_2$.

$$
\begin{aligned}
|f_t(\mathbf{u}) - f_t(\mathbf{v})| &= \left| \left(\mathbf{u}^T x^{(t)} - y^{(t)}\right)^2 - \left(\mathbf{v}^T x^{(t)} - y^{(t)}\right)^2 \right| && (33)\\
&= \left| \left(\mathbf{u}^T x^{(t)}\right)^2 + \left(y^{(t)}\right)^2 + 2y^{(t)}\mathbf{u}^T x^{(t)} - \left(\mathbf{v}^T x^{(t)}\right)^2 - \left(y^{(t)}\right)^2 - 2y^{(t)}\mathbf{v}^T x^{(t)} \right| && (34)\\
&= \left| \left(\mathbf{u}^T x^{(t)}\right)^2 - \left(\mathbf{v}^T x^{(t)}\right)^2 + 2y^{(t)}(\mathbf{u}-\mathbf{v})^T x^{(t)} \right| && (35)\\
&\leq \left| \left(\mathbf{u}^T x^{(t)}\right)^2 - \left(\mathbf{v}^T x^{(t)}\right)^2 \right| + 2\left|(\mathbf{u}-\mathbf{v})^T x^{(t)}\right| && (36)\\
&\leq \left| \left(\mathbf{u}^T x^{(t)}\right)^2 - \left(\mathbf{v}^T x^{(t)}\right)^2 \right| + 2\|\mathbf{u}-\mathbf{v}\|_2 \left|x^{(t)}\right| && (37)\\
&\leq \left(\mathbf{u}^T x^{(t)}\right)^2 + \left(\mathbf{v}^T x^{(t)}\right)^2 + 2 \cdot 2r \leq \left(\mathbf{u}^T x^{(t)}\right)^2 + 2 \cdot 2r = 2r^2 + 4r && (38)
\end{aligned}
$$

So, solving the inequality

$$2r^2 + 4r \leq L \cdot 2r \implies L \geq r + 2 \tag{39}$$

So, we have, by (29)

$$\textsc{Regret}(T) \leq \frac{1}{2\eta} r^2 + T\eta\,(r+2)^2 \tag{40}$$

We choose a suitable value of $\eta = \frac{1}{\sqrt{2T}}$ to minimize the regret bound. Then,

$$
\begin{aligned}
\textsc{Regret}(T) &\leq \frac{r^2}{2}\sqrt{2T} + \frac{T}{\sqrt{2T}}\,(r+2)^2 && (41)\\
&= \frac{r^2}{2}\sqrt{2T} + \frac{r^2}{2}\sqrt{2T} + 2r\sqrt{2T} + 2\sqrt{2T} && (42)\\
&= \sqrt{2T}\left(r^2 + 2r + 2\right) && (43)\\
&\leq \sqrt{2T}\left(r + \sqrt{2}\right)^2 = \mathcal{O}\left(\sqrt{T}\right) && (44)
\end{aligned}
$$

References

1. Lecture on Online Learning and Online Convex Optimization (2017, Dr. Akshay Krishnamurthy)
2. Lecture on Online Convex Optimization (Summer 2021, Dr. Thomas Kesselheim)
3. Upper bound on Scalar Product (Math Stack Exchange)
4. Lecture on No Regret Learning (Monsoon 2023, Dr. Syamantak Das)
5. Competitive Ratio vs Regret Minimization