Solving General LPs via MWU

April 12, 2023

1 Recap MWU

- Recall the MWU algorithm

Algorithm 1 The Multiplicative Weights Update (MWU) algorithm

MWU(ϵ):

- $w_i^1 \leftarrow 1 \quad \forall i$
- For $t = 1, \ldots, T$
  - Follows expert i with probability $p_i^t = \frac{w_i^t}{\sum_j w_j^t}$
  - After ℓ^t is revealed, $w_i^{t+1} \leftarrow w_i^t \cdot (1 - \epsilon \ell_i^t) \quad \forall i$

- $L_{\text{MWU}} := \sum_{t=1}^T \ell_{\text{MWU}}^t = \sum_{t=1}^T \langle p^t, \ell^t \rangle$ is our total loss
- $L_i := \sum_{t=1}^T \ell_i^t$ is the total loss of the expert i.

Theorem 1.1. *For $0 < \epsilon \le 1/2$, MWU(ϵ) guarantees that, for all i,*

$$L_{\text{MWU}} \le L_i + \epsilon T + \frac{\ln n}{\epsilon}.$$

- Let $L = (L_1, \ldots, L_n)$ be the list of the total loss of all experts.

- For any distribution of experts $z \in \Delta_n$, we have

$$L_{\text{MWU}} \leq \langle L, z \rangle + \epsilon T + \frac{\ln n}{\epsilon}.$$

2 The LP Feasibility Problem

- Let $\Delta_n = \{x \in \mathbb{R}^n_{\geq 0} \mid \mathbb{1}^\top x = 1\}$ be the set of probability distributions over n objects.
- Consider the following problem, given $A \in [-1, 1]^{m \times n}$ and $b \in \mathbb{R}^m$,

$$\begin{aligned} \text{Find} \quad & x \in \Delta_n \\ \text{s.t.} \quad & Ax \leq b. \end{aligned}$$

  or return that there is no feasible solution.

- **Exercise**:
  - If we can solve this problem, then we can solve any LP (Hint: binary search).
  - The solution will not be exact, but we can get arbitrarily close to optimal solution.
- **Today:**
  - solve a relaxed version of the above problem:

$$\begin{aligned} \text{Find} \quad & x \in \Delta_n \\ \text{s.t.} \quad & Ax \leq b + 2\epsilon \mathbb{1}. \end{aligned}$$

    or return that there is no feasible solution $x \in \Delta_n$ where $Ax \leq b$.
  - **Time:** $O(\text{nnz}(A) \ln(n)/\epsilon^2)$.

3 The Whack-a-Mole Algorithm

- **Natural Idea**:
  - Keep "fixing" a violated constraint.
  - More precisely, how to fix? use MWU

Algorithm 2 Whack-a-Mole on General LPs

- Initialize $w^1 \leftarrow \mathbb{1} \in \mathbb{R}^n$
- Maintain $x^t = w^t / W^t$ at all time where $W^t = \sum_{j=1}^n w_j^t$.
- **For** $t = 1, \ldots, T$ where $T = \ln(n)/\epsilon^2$
  - **If** $Ax^t \leq b + 2\epsilon\mathbb{1}$, **return** $x = x^t$.
  - **Else** there is a constraint $i_t \in [m]$ where $A_{(i_t,\cdot)} x^t > b + 2\epsilon$,
    - $w_j^{t+1} \leftarrow w_j^t \cdot (1 - A_{i_t j}) \forall j \in [n]$. ("**whack constraint** i_t")
- **Return** "no feasible solution".

- Running time: $O(\mathrm{nnz}(A) \ln(n)/\epsilon^2)$.

Lemma 3.1. *Suppose there exists a feasible solution x^*. Then, the algorithm must return $x \in \Delta_n$ where $Ax \leq b + 2\epsilon\mathbb{1}$.*

- Suppose for contradiction that after T iterations, x is not returned even if x^* exists.
- Observe: the Whack-a-Mole algorithm implements MWU(ϵ):
  - experts = variables
  - We choose a distribution $x^t = w^t / W^t \in \mathbb{R}^n_{\geq 0}$ on experts.
  - Get a loss vector of experts

$$\ell^t = A_{(i_t,\cdot)} \in [-1, 1]^n$$

- Then, update the weights

$$w_j \leftarrow w_j \cdot (1 - \epsilon \ell_j^t) \forall j \in [n]$$

exactly as in MWU(ϵ).

- For each day t,
  - The loss of MWU is $\langle \ell^t, x^t \rangle = A_{(i_t,\cdot)} x^t > b_{i_t} + 2\epsilon$
  - But we know that $\langle \ell^t, x^* \rangle = A_{(i_t,\cdot)} x^* \leq b_{i_t}$ as x^* is feasible.
- Summing over all days, we have

$$L_{\text{MWU}} - \langle L, x^* \rangle = \sum_{t=1}^{T} \left\langle \ell^t, x^t \right\rangle - \left\langle \ell^t, x^* \right\rangle > 2\epsilon T$$

- On the other hand, the no-regret bound implies

$$L_{\text{MWU}} - \langle L, x^* \rangle \leq \epsilon T + \frac{\ln n}{\epsilon} \leq 2\epsilon T$$

because $T = \frac{\ln n}{\epsilon^2}$.

- This is a contradiction.

4 Removing Strong Promise

Exercise 4.1. Given the above algorithm, suppose we only promise that

- $A \in [-\rho, \rho]^{m \times n}$ and
- $x \in \tau \cdot \Delta_n$ (i.e. $\sum_i x_i \leq \tau$ and $x_i \geq 0$).

Then we can in $O(\mathrm{nnz}(A)\ln(n)(\frac{\rho\tau}{\epsilon})^2)$ time

$$
\begin{aligned}
\text{Find} \quad & x \in \tau \cdot \Delta_n \\
\text{s.t.} \quad & Ax \leq b + 2\epsilon \mathbf{1}.
\end{aligned}
$$

or return that there is no feasible solution $x \in \tau \cdot \Delta_n$ where $Ax \leq b$.

Hint: just scaling and set ϵ as $\epsilon/\tau\rho$.

- This is not great: ρ, τ can be huge.
- The technique for removing this dependency: *width-independent MWU*
  - Not in this class.
  - Work for some special classes of LP: when all entries of LPs are non-negative.