CS655 Reading assignment - 2
Multiplicative Weights Algorithm for Vectors and Matrices (Chapters 2 and 3 of Thesis on Efficient Algorithms using Multiplicative Weights Method by Satyen Kale)

apanda and ananya20

November 2023

1 The Problem: Vectors

1.1 The basic idea

Let us say we want to invest in an arbitrary stock, and we know that it can either go up and give us a payoff of +1, or go down and give us a payoff -1. However, we can only predict the nature of the stock and invest accordingly, so we find out only at a later time whether our payoff is +1 or -1. We are allowed to predict the stocks' performance for timestamps $t = 1 \cdots T$, and then receive our total payoff at the end.
Since it's a matter of our future, we are allowed to consult n "experts", who make predictions about our favorite stock at every timestamp for the next timestamp. Of course, the payoff rule does not change for them, and it is expected that the prediction of some of the experts will be proven wrong. The catch is that, if we knew what the total accumulated payoff (at time $T + 1$) would be for each of those experts, we would simply have followed the strategy for the expert with the best payoff. But we only know the payoffs' for all the experts after the timestamp t has passed.

1.2 The basic strategy: Weighted Majority Algorithm

The Weighted Majority Algorithm (proposed by Littlestone and Warmuth) goes like this for our scenario:

Algorithm 1 The Weighted Majority Algorithm for The Stock Scenario

```
for ith expert from the n experts in the game do
    Initialise a weight w_i^(1) = 1.
end for
Assign an ε ∈ (0, 1/2]
for t = 1 ··· T do
    Pick the weighted majority voting from the experts. Say if the majority weightage goes for +1, choose +1.
    Break ties arbitrarily.
    Wait for the payoff scores to arrive for you and for the experts.
    for ith expert in n experts do
        if i did not make the correct prediction then
            w_i^(t+1) = (1 − ε) w_i^(t)
        else
            continue
        end if
    end for
end for
```

Definition 1.1. $m^{(t)}$: number of wrong predictions our algorithm makes until time t
$m_i^{(t)}$: number of wrong predictions ith expert makes until time t

Definition 1.2. $\Phi^{(t)} = \sum_{i=1}^{n} w_i{}^{(t)}$

Theorem 1. *After time T, if $m_i^{(T)}$ is the number of mistakes ith expert made, and $m^{(T)}$ is the number of mistakes we made, then $\forall i \in n$ experts*

$$m^{(T)} \leq \frac{2 \ln n}{\varepsilon} + 2(1 + \varepsilon) m_i^{(T)}$$

Remark. If the above is true $\forall n$ experts, then it must be true for the expert who made it to time T with the best payoff and the least mistakes.

Proof.

$$
\begin{aligned}
w_i^{(T+1)} &= \prod_{k=1}^{m_i^{(T)}} (1-\varepsilon) \prod_{k'=1}^{T-m_i^{(T)}} 1 \\
&= (1-\varepsilon)^{m_i^{(T)}}
\end{aligned}
$$

We know that $\Phi^{(1)} = n$. We also know that if in our algorithm, we made the wrong prediction, it implies that at least half of the n experts made the wrong prediction. Thus, at least half of the weights will decrease by a factor of $1-\varepsilon$

$$
\begin{aligned}
\Phi^{(t+1)} &\leq \Phi^{(t)}(\frac{1}{2} + (1-\varepsilon)\frac{1}{2}) \\
&= \Phi^{(t)}(1-\varepsilon/2)
\end{aligned}
$$

This factor will be multiplied to $\Phi^{(t)}$ only for times our algorithm makes a mistake, thus by induction

$$
\begin{aligned}
\Phi^{(T+1)} &\leq \Phi^{(1)}(1-\varepsilon/2)^{m^{(T)}} \\
&= n(1-\varepsilon/2)^{m^{(T)}}
\end{aligned}
$$

Using the fact that $\Phi^{(t)} \geq w_i^{(t)}\ \forall i$, we can see that

$$
\begin{aligned}
\Phi^{(T+1)} &\geq w_i^{(T+1)} \\
n(1-\varepsilon/2)^{m^{(T)}} &\geq (1-\varepsilon)^{m_i^{(T)}}
\end{aligned}
$$

Taking natural logarithm on both sides

$$
\begin{aligned}
\ln n + m^{(T)} \ln(1-\varepsilon/2) &\geq m_i^{(T)} \ln(1-\varepsilon) \\
-m^{(T)} \ln(1-\varepsilon/2) &\leq \ln n - m_i^{(T)} \ln(1-\varepsilon)
\end{aligned}
$$

Since $\varepsilon \leq 1/2$, $-\ln(1-\varepsilon) \leq \varepsilon + \varepsilon^2$

$$
\therefore m^{(T)}(\varepsilon/2 + \varepsilon^2/4) \leq \ln n + m_i^{(T)}(\varepsilon + \varepsilon^2)
$$

Neglecting $\varepsilon^2/4$, and dividing by $\varepsilon/2$, we arrive at the result. □

1.3 The Multiplicative Weights Algorithm

This algorithm is a more "generalised" version of the Weighted Majority algorithm, in a way as now the "prediction" can have infinite choices i.e. say we have to now predict the price of the stock instead of just it's direction, or maybe we are just looking for a course of action to proceed in investments. The number of experts remain n.
Let us now take a probability distribution over the experts, expressed as vector $\mathbf{p}^{(t)}$. Let us say that for whatever course of action (or price or choice) the expert i recommends, we get a cost (*costs, not payoff. In case we get a profit instead of cost, we can put it as negative cost.*) $m_i^{(t)}$, which altogether for n experts is the vector $\mathbf{m}^{(t)}$. In order to approach the algorithm and subsequent proofs, Kale also makes the assumption in his thesis that all $m_i^{(t)} \in [-1, 1]$. The expected cost of a timestamp t, $= \mathbf{m}^{(t)} \cdot \mathbf{p}^{(t)}$. Over timestamps $t = 1 \cdots T$, the expected cost is $\sum_{t=t}^{T} \mathbf{m}^{(t)} \cdot \mathbf{p}^{(t)}$. This is what we must minimise, and bring as close as possible to the minimum cost (*minimum since this is a cost*), had we known all costs before-hand, i.e. $\min_i \sum_{i=1}^{n} m_i^{(t)}$.
Intuitively, if an expert made lesser cost than the others, for the next timestamp we must reward the expert with increased probability so that they are more likely to get picked next time. And for experts who incurred high costs, their probability has to be decreased to try and avoid further costs in later rounds. This is exactly what the Multiplicative Weights algorithm does.

Algorithm 2 The Multiplicative Weights Algorithm

for ith expert from the n experts in the game **do**
  Initialise a weight $w_i^{(1)} = 1$.
end for
Assign an $\varepsilon \in (0, \frac{1}{2}]$
for $t = 1 \cdots T$ **do**
  Assign probability distribution vector $\mathbf{p}^{(t)} = \{\frac{w_i^{(t)}}{\Phi^{(t)}}\}$ $\forall i \in 1 \cdots n$ and $\Phi^{(t)} = \sum_{i=1}^{n} w_i^{(t)}$.
  Pick an expert from the probability distribution, run the course of action and get the cost vector $\mathbf{m}^{(t)}$.
  for ith expert in n experts **do**
    if $m_i^{(t)} \geq 0$ **then**
      $w_i^{(t+1)} = w_i^{(t)}(1-\varepsilon)^{m_i^{(t)}}$
    else
      $w_i^{(t+1)} = w_i^{(t)}(1+\varepsilon)^{-m_i^{(t)}}$
    end if
  end for
end for

As before, we get an upper bound on $\sum_{t=1}^{T} \mathbf{m}^{(t)} \cdot \mathbf{p}^{(t)}$.

Theorem 2. *After T timestamps,*

$$\sum_{t=1}^{T} \mathbf{m}^{(t)} \cdot \mathbf{p}^{(t)} \leq \frac{\ln n}{\varepsilon} + \sum_{t=1}^{T} m_i^{(t)} + \varepsilon \sum_{t=1}^{T} |m_i^{(t)}| \qquad \forall i \in 1 \cdots n$$

Definition 1.3. The notation $\sum_{i:m_i^{(t)}<0} m_i^{(t)}$ means the summation of all $m_i^{(t)}$ over i given t such that $m_i^{(t)} < 0$. Similar explanation follows for $\sum_{i:m_i^{(t)}\geq 0} m_i^{(t)}$.
Notation $\sum_{<0} m_i^{(t)}$ implies the summation over all timestamps t for which $m_i^{(t)} < 0$. Similar explanation follows for $\sum_{\geq 0} m_i^{(t)}$.
Proof.

$$\begin{aligned}
\Phi^{(t+1)} &= \sum_{i=1}^{n} w_i^{(t+1)} \\
&= \sum_{i:m_i^{(t)}\geq 0} w_i^{(t)}(1-\varepsilon)^{m_i^{(t)}} + \sum_{i:m_i^{(t)}<0} w_i^{(t)}(1+\varepsilon)^{-m_i^{(t)}} \\
&\leq \sum_{i:m_i^{(t)}\geq 0} w_i^{(t)}(1-m_i^{(t)}\varepsilon) + \sum_{i:m_i^{(t)}<0} w_i^{(t)}(1-m_i^{(t)}\varepsilon) \qquad (3) \\
&= \sum_{i=1}^{n} w_i^{(t)}(1-m_i^{(t)}\varepsilon) \qquad (4)
\end{aligned}$$

Remark. In the presentation, a query about the "+" sign for ε the weight in case $m_i^{(t)} < 0$ was made. We attempt to address the same here. The result from step (3) (and subsequently step (4)) was reached only because of the convex nature of expial functions that gives the properties

$$(1-\varepsilon)^x \leq (1-\varepsilon x) \qquad \forall x \in [0,1]$$
$$(1+\varepsilon)^{-x} \leq (1-\varepsilon x) \qquad \forall x \in [-1,0]$$

It may also be noted that this was also the reason why the assumption that $m_i^{(t)} \in [-1,1]$ was made.

$$\begin{aligned}
\Phi^{(t+1)} &\leq \Phi^{(t)} \sum_{i=1}^{n} w_i^{(t)} \Phi^{(t)} (1-m_i^{(t)}\varepsilon) \\
&= \Phi^{(t)} - \varepsilon \Phi^{(t)} \sum_{i=1}^{n} m_i^{(t)} p_i^{(t)} \\
&= \Phi^{(t)}(1-\varepsilon \mathbf{m}^{(t)} \cdot \mathbf{p}^{(t)}) \\
&\leq \Phi^{(t)} \exp(-\varepsilon \mathbf{m}^{(t)} \cdot \mathbf{p}^{(t)}) \qquad (\because (1-x) \leq e^{-x})
\end{aligned}$$

Thus after T timestamps,

$$\begin{aligned}
\Phi^{(T+1)} &\leq \Phi^{(T)} \exp\left(-\varepsilon \mathbf{m}^{(T)} \cdot \mathbf{p}^{(T)}\right) \\
&\leq \Phi^{(T-1)} \exp\left(-\varepsilon \mathbf{m}^{(T-1)} \cdot \mathbf{p}^{(T-1)}\right) \exp\left(-\varepsilon \mathbf{m}^{(T)} \cdot \mathbf{p}^{(T)}\right) \\
&\qquad\qquad \vdots \\
&\leq \Phi^{(1)} \exp\left(-\varepsilon \sum_{t=1}^{T} \mathbf{m}^{(t)} \cdot \mathbf{p}^{(t)}\right) \\
&= n \exp\left(-\varepsilon \sum_{t=1}^{T} \mathbf{m}^{(t)} \cdot \mathbf{p}^{(t)}\right)
\end{aligned}$$

We also know that $\Phi^{(T+1)} \geq w_i^{(T+1)} = \left[(1-\varepsilon)^{\sum_{\geq 0} m_i^{(t)}}\right] \cdot \left[(1+\varepsilon)^{-\sum_{<0} m_i^{(t)}}\right]$ Therefore,

$$\begin{aligned}
\left[(1-\varepsilon)^{\sum_{\geq 0} m_i^{(t)}}\right] \cdot \left[(1+\varepsilon)^{-\sum_{<0} m_i^{(t)}}\right] &\leq n \exp\left(-\varepsilon \sum_{t=1}^{T} \mathbf{m}^{(t)} \cdot \mathbf{p}^{(t)}\right) \\
\sum_{\geq 0} m_i^{(t)} \ln\left[(1-\varepsilon)\right] - \sum_{<0} m_i^{(t)} \ln\left[(1+\varepsilon)\right] &\leq \ln n - \varepsilon \sum_{t=1}^{T} \mathbf{m}^{(t)} \cdot \mathbf{p}^{(t)} \\
\sum_{t=1}^{T} \mathbf{m}^{(t)} \cdot \mathbf{p}^{(t)} &\leq \frac{\sum_{<0} m_i^{(t)} \ln\left[(1+\varepsilon)\right]}{\varepsilon} - \frac{\sum_{\geq 0} m_i^{(t)} \ln\left[(1-\varepsilon)\right]}{\varepsilon} + \frac{\ln n}{\varepsilon}
\end{aligned}$$

We also know that for $\varepsilon \leq \frac{1}{2}$,

$$-\ln\left[1-\varepsilon\right] \leq \varepsilon + \varepsilon^2$$

and,

$$\ln\left[1+\varepsilon\right] \geq \varepsilon - \varepsilon^2$$

So the inequation can also be written as,

$$\begin{aligned}
\sum_{t=1}^{T} \mathbf{m}^{(t)} \cdot \mathbf{p}^{(t)} &\leq \frac{\ln n}{\varepsilon} + \sum_{<0} m_i^{(t)} \left[1-\varepsilon\right] + \sum_{\geq 0} m_i^{(t)} \left[1+\varepsilon\right] \\
&= \frac{\ln n}{\varepsilon} + \sum_{t=1}^{T} m_i^{(t)} + \varepsilon\left(\sum_{\geq 0} m_i^{(t)} - \sum_{<0} m_i^{(t)}\right) \\
&= \frac{\ln n}{\varepsilon} + \sum_{t=1}^{T} m_i^{(t)} + \sum_{t=1}^{T} |m_i^{(t)}|
\end{aligned}$$

□

1.4 Applications

- **Approximate Solution of Zero-Sum games**
- **Approximate Solution of Linear Programs in the convex domain** which leads to applications in
  - Fractional Covering Problems
  - Fractional Packing Problems

2 The Problem: Matrices

2.1 How and Why Matrices?

The vector problem we discussed above can be generalised to matrices. The standard basis vectors $\mathbf{e}_1 \cdots \mathbf{e}_n$ can be called the n experts, with $\mathbf{m}^{(t)} = \text{diag}[\mathbf{M}^{(t)}]$ where $\mathbf{M}^{(t)}$ is now the cost matrix. The probability distribution, similarly can be expanded to the probability density matrix $\mathbf{P}^{(t)}$ such that $\mathbf{p}^{(t)} = \text{diag}[\mathbf{P}^{(t)}]$.
The reason a generalisation to matrices (not just diagonal matrices or standard basis vectors) is great is because it helps not only get a better bound on the number of mistakes the Multiplicative Weights algorithm makes, but also because it has applications in solving semidefinite programs.

2.2 Definitions and Algorithm

Definition 2.1.

$$\exp{(A)} = \sum_{i=0}^{\infty} \frac{\mathbf{A}^i}{i}$$

for a matrix $\mathbf{A}$

Definition 2.2.

$$\mathbf{A} \bullet \mathbf{B} = \mathbf{Tr}(\mathbf{AB})$$

Theorem 3. *Golden-Thompson inequality: for any two real matrices* $\mathbf{A}$*,* $\mathbf{B}$*,*

$$\mathbf{Tr}(\exp{(\mathbf{A}+\mathbf{B})}) \leq \ \mathbf{Tr}(\exp{(\mathbf{A})}\exp{(\mathbf{B})})$$

Corollary 3.1. *Let* $\varepsilon_1 = 1 - e^{-\varepsilon}$*,* $\varepsilon_2 = e^{\varepsilon} - 1$*, then:*

- *If all eigenvalues of a matrix* $\mathbf{A}$ *lie in* $[0, 1]$*, then* $\exp{(-\varepsilon\mathbf{A})} \preceq \mathbf{I}_n - \varepsilon_1\mathbf{A}$
- *If all eigenvalues of a matrix* $\mathbf{A}$ *lie in* $[-1, 0]$*, then* $\exp{(-\varepsilon\mathbf{A})} \preceq \mathbf{I}_n - \varepsilon_2\mathbf{A}$

Now our experts are infinite in number, because we will be considering vectors from the unit sphere in n dimensions ie. $\mathbb{S}^{n-1}$. We also have a cost matrix $\mathbf{M}^{(t)} \in \mathbb{R}^{n\times n}$ such that, for a given vector $\mathbf{v} \in \mathbb{S}^{n-1}$, the cost of picking that vector's "course of action" at timestamp t will be $\mathbf{v}^\top\mathbf{M}^{(t)}\mathbf{v}$, and that this cost will be $\in [-1, 1]$. It may be noted that this definition is in sync to the vector-to-matrix generalisation we tried to give in the previous subsection.
We also now take a probability distribution $\mathcal{D}$ over the unit sphere. Thus,

$$\mathbb{E}_{\mathbf{v}\in\mathcal{D}}[\mathbf{v}^\top\mathbf{M}^{(t)}\mathbf{v}] = \mathbb{E}_{\mathbf{v}\in\mathcal{D}}[\mathbf{M}^{(t)} \bullet \mathbf{v}]v^\top = \mathbf{M}^{(t)} \bullet \mathbb{E}_{\mathbf{v}\in\mathcal{D}}[\mathbf{v}\mathbf{v}^\top]$$

Let $\mathbf{P}^{(t)} = \mathbb{E}_{\mathbf{v}\in\mathcal{D}}[\mathbf{v} \cdot \mathbf{v}^\top]$. It may be noted that since for any arbitrary $\mathbf{u} \in \mathbb{R}^n$,

$$\mathbf{u}^\top\mathbf{P}^{(t)}\mathbf{u} = \mathbf{u}^\top\mathbb{E}_{\mathbf{v}\in\mathcal{D}}[\mathbf{v}\mathbf{v}^\top]\mathbf{u} = \mathbb{E}_{\mathbf{v}\in\mathcal{D}}[\mathbf{u}^\top\mathbf{v}\mathbf{v}^\top\mathbf{u}] = \mathbb{E}_{\mathbf{v}\in\mathcal{D}}[(\mathbf{u}\mathbf{v}^\top)^\top(\mathbf{u}\mathbf{v}^\top)] \geq 0$$

$\mathbf{P}^{(t)}$ is positive semidefinite. Moreover because all $\mathbf{v} \in \mathcal{D}$ are unit vectors $\Rightarrow \mathbf{Tr}(\mathbf{P}^{(t)}) = \mathbf{Tr}(\mathbf{v}\mathbf{v}^\top) = 1$, $\mathbf{P}^{(t)}$ can be called a density matrix for the distribution $\mathcal{D}$.
Now we know our goal is to minimise the expected cost, i.e. $\Sigma_{t=1}^{T}\mathbf{M}^{(t)} \bullet \mathbf{P}^{(t)}$. By Rayleigh-Ritz method, the minimum value comes out to be $\lambda_n(\Sigma_{t=1}^{T}\mathbf{M}^{(t)})$, the smallest eigenvalue of the matrix $\Sigma_{t=1}^{T}\mathbf{M}^{(t)}$.

Algorithm 3 The Matrix Multiplicative Weights Algorithm

1: Initialise a weight matrix $\mathbf{W}^{(t)} = \mathbf{I}_n$.
2: Assign an $\varepsilon \in (0, \frac{1}{2}]$
3: **for** $t = 1 \cdots T$ **do**
4: Assign density matrix $\mathbf{P}^{(t)} = \frac{\mathbf{W}^{(t)}}{\mathbf{Tr}(\mathbf{W}^{(t)})}$.
5: Pick an expert from the probability distribution $\mathcal{D}$ (as specified by $\mathbf{P}^{(t)}$, run the course of action and get the cost matrix $\mathbf{M}^{(t)}$.
6: Update the weight matrix as $\mathbf{W}^{(t+1)} = \exp{(-\varepsilon\sum_{\tau=1}^{t}\mathbf{M}^{(\tau)})}$
7: **end for**

Definition 2.3. $\sum_{\succeq\mathbf{0}}\mathbf{M}^{(t)}$ implies the summation of $\mathbf{M}^{(t)}$ over all t when $\mathbf{M}^{(t)} \succeq \mathbf{0}$. Similar definition for $\sum_{\preceq\mathbf{0}}\mathbf{M}^{(t)}$

Theorem 4. *After* T *rounds, for any expert* $\mathbf{v}$ *as specified in the setting,*

$$(1-\varepsilon)\sum_{\succeq\mathbf{0}}\mathbf{M}^{(t)} \bullet \mathbf{P}^{(t)} + [(1+\varepsilon)\sum_{\preceq\mathbf{0}}\mathbf{M}^{(t)} \bullet \mathbf{P}^{(t)} \leq \sum_{t=1}^{T}\mathbf{v}^\top\mathbf{M}^t\mathbf{v} + \frac{\ln n}{\varepsilon}$$

Proof. We take $\Phi^{(t)} = \mathbf{Tr}(\mathbf{W}^{(t)})$. We know that,

$$
\begin{aligned}
\Phi^{(t+1)} &= \mathbf{Tr}(\mathbf{W}^{(t+1)}) \\
&= \mathbf{Tr}(\exp(-\varepsilon \sum_{\tau=1}^{t} \mathbf{M}^{(\tau)})) \\
&\leq \mathbf{Tr}(\exp(-\varepsilon \sum_{\tau=1}^{t-1} \mathbf{M}^{(\tau)}) \exp(-\varepsilon \mathbf{M}^{(t)})) \qquad (\because \text{Golden-Thompson's inequality}) \\
&= \mathbf{W}^{(t)} \bullet \exp(-\varepsilon \mathbf{M}^{(t)}) \\
&\leq \begin{cases} \mathbf{W}^{(t)} \bullet (\mathbf{I} - \varepsilon_1 \mathbf{M}^{(t)}) & \text{if } \mathbf{M}^{(t)} \succeq \mathbf{0} \\ \mathbf{W}^{(t)} \bullet (\mathbf{I} - \varepsilon_2 \mathbf{M}^{(t)}) & \text{if } \mathbf{M}^{(t)} \preceq \mathbf{0} \end{cases} \qquad (\text{From Corollary 3.1}) \\
&= \begin{cases} \mathbf{Tr}(\mathbf{W}^{(t)}) \cdot (1 - \varepsilon_1 \mathbf{M}^{(t)} \bullet \mathbf{P}^{(t)}) & \text{if } \mathbf{M}^{(t)} \succeq \mathbf{0} \\ \mathbf{Tr}(\mathbf{W}^{(t)}) \cdot (1 - \varepsilon_2 \mathbf{M}^{(t)} \bullet \mathbf{P}^{(t)}) & \text{if } \mathbf{M}^{(t)} \preceq \mathbf{0} \end{cases} \\
&\leq \begin{cases} \Phi^{(t)} \cdot \exp(-\varepsilon_1 \mathbf{M}^{(t)} \bullet \mathbf{P}^{(t)}) & \text{if } \mathbf{M}^{(t)} \succeq \mathbf{0} \\ \Phi^{(t)} \cdot \exp(-\varepsilon_2 \mathbf{M}^{(t)} \bullet \mathbf{P}^{(t)}) & \text{if } \mathbf{M}^{(t)} \preceq \mathbf{0} \end{cases}
\end{aligned}
$$

Now, as before, by induction, and because $\Phi^{(1)} = \mathbf{Tr}(\mathbf{W}^{(1)}) = \mathbf{Tr}(\mathbf{I}_n) = n$,

$$
\Phi^{(T+1)} \leq n \exp(-\varepsilon_1 \sum_{\succeq \mathbf{0}} \mathbf{M}^{(t)} \bullet \mathbf{P}^{(t)} - \varepsilon_2 \sum_{\preceq \mathbf{0}} \mathbf{M}^{(t)} \bullet \mathbf{P}^{(t)})
$$

We also have

$$
\Phi^{(T+1)} = \mathbf{Tr}(\mathbf{W}^{(T+1)}) = \mathbf{Tr}(\exp(-\varepsilon \sum_{\tau=1}^{t} \mathbf{M}^{(\tau)})) \geq \exp(-\varepsilon \lambda_n (\Sigma_{t=1}^{T} \mathbf{M}^{(t)})) \tag{3.1}
$$

. Thus,

$$
\exp(-\varepsilon \lambda_n (\Sigma_{t=1}^{T} \mathbf{M}^{(t)})) \leq n \exp(-\varepsilon_1 \sum_{\succeq \mathbf{0}} \mathbf{M}^{(t)} \bullet \mathbf{P}^{(t)} - ep_2 \sum_{\preceq \mathbf{0}} \mathbf{M}^{(t)} \bullet \mathbf{P}^{(t)})
$$

We used sum facts to come to the theorem:

- $\varepsilon_1 \leq \varepsilon - \varepsilon^2$ and $\varepsilon_2 \leq \varepsilon + \varepsilon^2$ $\forall \varepsilon \geq 1/2$.
- $\lambda_n (\Sigma_{t=1}^{T} \mathbf{M}^{(t)}) \leq \sum_{t=1}^{T} \mathbf{v}^\top \mathbf{M}^{(t)} \mathbf{v}$.

□

Remark. For (3.1), we used,

$$
\mathbf{Tr}(e^{\mathbf{A}}) \geq \sum_{k=1}^{n} e^{\lambda_k(\mathbf{A})} \geq e^{\lambda_n(\mathbf{A})}
$$

Theorem 5: A theorem for Linear Programs

A result for the approach to linear programs described in the vector section

Assumptions

- Let $\delta > 0$ be a small error margin we are willing to tolerate.
- We have access to an (ϵ, δ)-bounded Oracle which assists in finding solutions or confirming the absence thereof within these error bounds.
- We aim to find an x such that $Ax \leq b + \delta \cdot \mathbf{1}$, where $\mathbf{1}$ is a vector of ones.

The Oracle is queried with a series of weights, $w^{(t)}$, which are adjusted in each round based on the deviation of Ax from b. The weights are updated using the rule:

$$
m^{(t)}(p) = \frac{1}{2\epsilon} |Ax^{(t)} - b|,
$$

where $x^{(t)}$ is the solution proposed by the Oracle in round t. The algorithm ensures that the expected cost in each round:

$$m^{(t)} \cdot p^{(t)} = \frac{1}{2\epsilon}|Ax^{(t)} - b| \cdot p^{(t)} \geq \frac{1}{2\epsilon}p^{(t)}(Ax^{(t)} - b) \geq 0.$$

After T rounds, we ensure that:

$$\sum_{t=1}^{T} p^{(t)}(Ax^{(t)} - b) \leq \frac{\ln m}{\epsilon} + \delta T,$$

leading us to conclude that if a solution exists within the given error margin, the algorithm will find it or otherwise declare the system infeasible.

Performance

- The algorithm calls the Oracle $O(\frac{\log m}{\delta^2})$ times, which is a manageable number of iterations.
- Each Oracle call takes $O(m)$ time, where m is the number of constraints.