Lecture 7. Prediction with Expert Advice

Advanced Optimization (Fall 2022)

Peng Zhao

zhaop@lamda.nju.edu.cn

Nanjing University

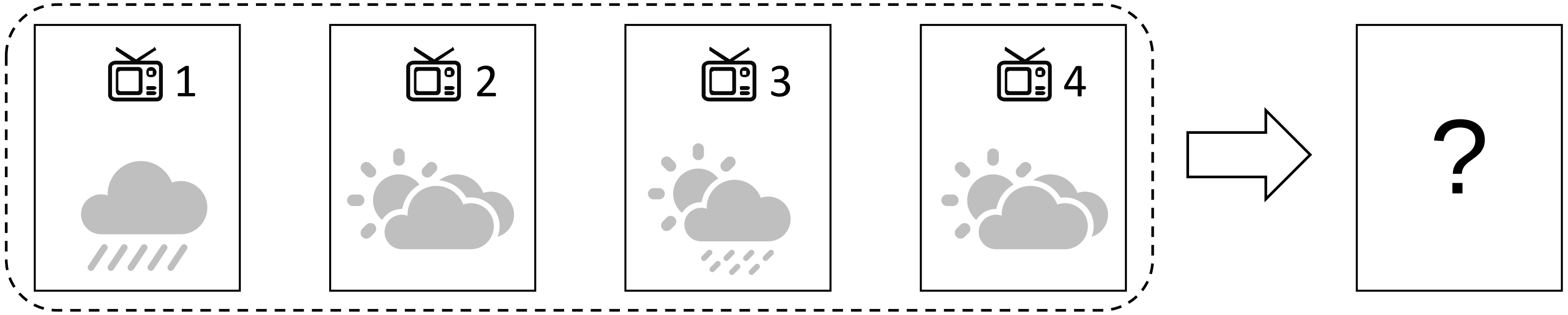
Outline

- PEA Problem Formulation
- Hedge Algorithm
- Upper and Lower Bounds
- OGD and Hedge

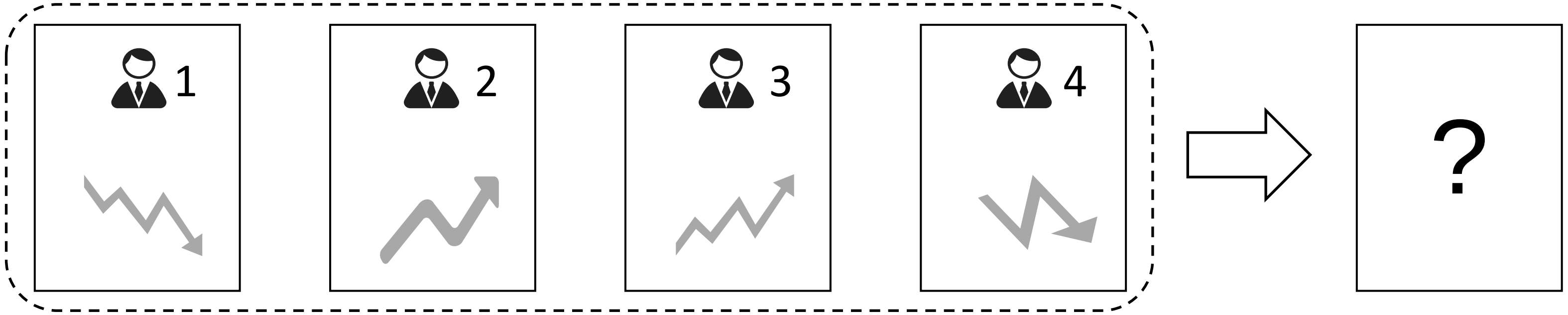
Prediction with Expert Advice

- A ubiquitous problem in real life:

Weather report:



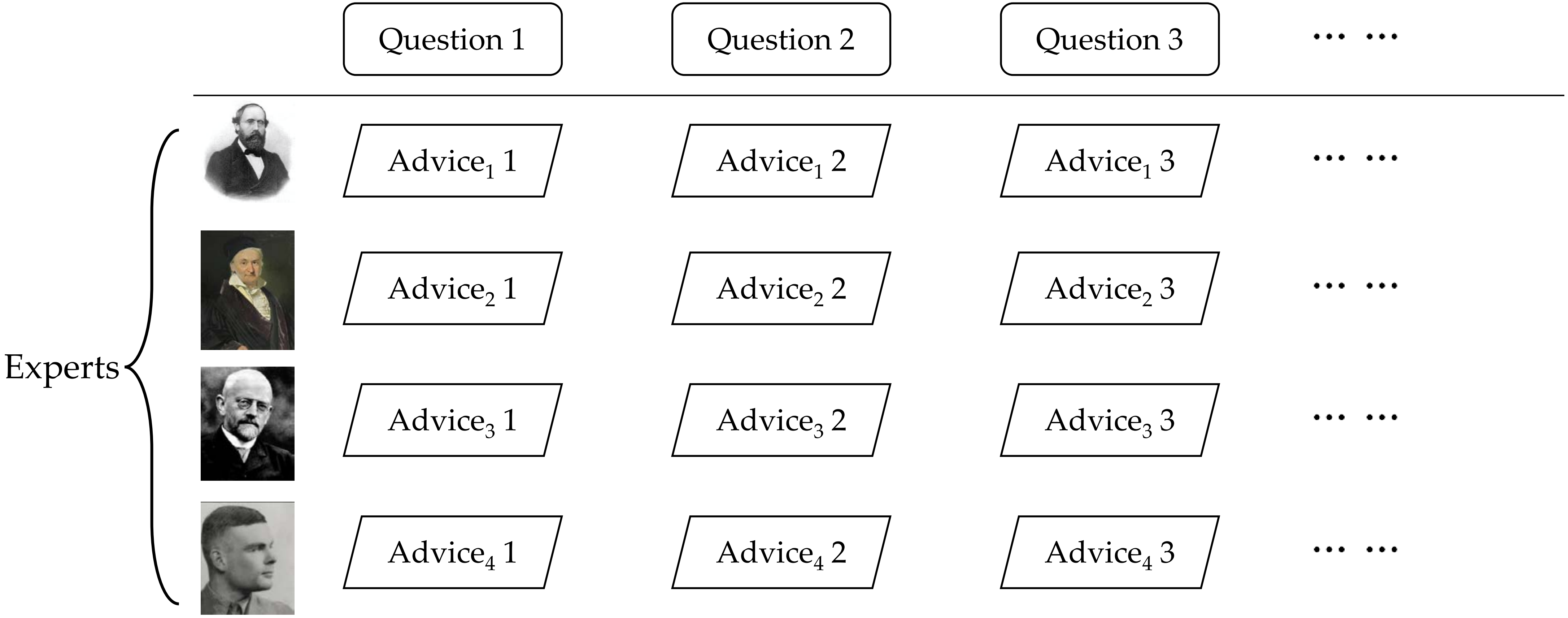
Stock market prediction:



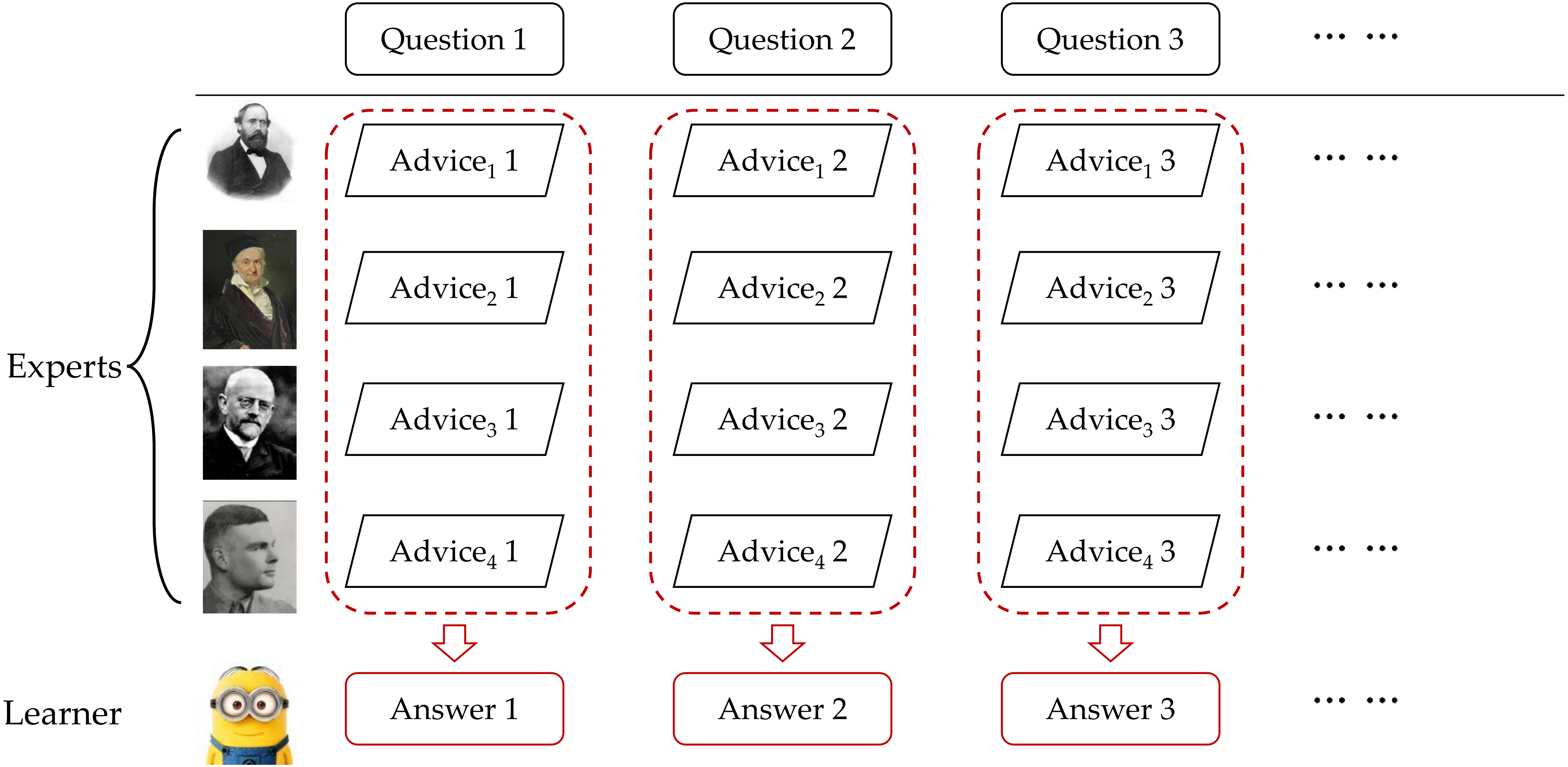
PEA Problem Setup

Question 1 | Question 2 | Question 3 | … …

PEA Problem Setup



PEA Problem Setup



PEA Problem Formulization

At each round $t = 1, 2, \cdots$

(1) the player first picks a weight $\boldsymbol{p}_t$ from a simplex Δ_N;

(2) and simultaneously environments pick a loss vector $\boldsymbol{\ell}_t \in \mathbb{R}^N$;

(3) the player suffers loss $f_t(\boldsymbol{p}_t) \triangleq \langle \boldsymbol{p}_t, \boldsymbol{\ell}_t \rangle$, observes $\boldsymbol{\ell}_t$ and updates the model.

- We typically assume that $\forall t \in [T]$ and $i \in [N]$, $0 \leq \boldsymbol{\ell}_t(i) \leq 1$.
- Make our prediction by combining N experts' advice.

PEA Problem Formulization

At each round $t = 1, 2, \cdots$

(1) the player first picks a weight $\boldsymbol{p}_t$ from a simplex Δ_N;

(2) and simultaneously environments pick a loss vector $\boldsymbol{\ell}_t \in \mathbb{R}^N$;

(3) the player suffers loss $f_t(\boldsymbol{p}_t) \triangleq \langle \boldsymbol{p}_t, \boldsymbol{\ell}_t \rangle$, observes $\boldsymbol{\ell}_t$ and updates the model.

- The goal is to minimize the regret with respect to the best expert:

$$\text{Regret}_T = \sum_{t=1}^{T} \langle \boldsymbol{p}_t, \boldsymbol{\ell}_t \rangle - \min_{\boldsymbol{p} \in \Delta_N} \sum_{t=1}^{T} \langle \boldsymbol{p}, \boldsymbol{\ell}_t \rangle = \sum_{t=1}^{T} \langle \boldsymbol{p}_t, \boldsymbol{\ell}_t \rangle - \min_{i \in [N]} \sum_{t=1}^{T} \boldsymbol{\ell}_t(i)$$

A Natural Solution for PEA

- Follow the Leader (FTL)

  Idea: Select the expert that performs best so far, namely,

$$\boldsymbol{p}_t^{\mathrm{FTL}} = \operatorname*{arg\,min}_{\boldsymbol{p} \in \Delta_N} \langle \boldsymbol{p}, L_{t-1} \rangle$$

  where $L_{t-1} \triangleq \sum_{s=1}^{t-1} \boldsymbol{\ell}_s \in \mathbb{R}^N$ is the cumulative loss vector.

A Natural Solution for PEA

- However, FTL may achieve ***linear regret*** in the worst case!

$\ell_1(1) = 0.49$ ⇨ $\ell_2(1) = 1$ ⇨ $\ell_3(1) = 0$ ⋯ ⋯

$\ell_1(2) = 0.51$ ⇨ $\ell_2(2) = 0$ ⇨ $\ell_3(2) = 1$ ⋯ ⋯

$$\text{Regret}_T = \sum_{t=1}^{T} \langle \boldsymbol{p}_t, \boldsymbol{\ell}_t \rangle - \min_{i \in [N]} \sum_{t=1}^{T} \boldsymbol{\ell}_t(i)$$

$$= T - \frac{T}{2} = \mathcal{O}(T)$$

A Natural Solution for PEA

- Follow the Leader (FTL)

  Idea: Select the expert that performs best so far, namely,

$$\boldsymbol{p}_t^{\text{FTL}} = \mathop{\arg\min}_{\boldsymbol{p} \in \Delta_N} \langle \boldsymbol{p}, L_{t-1} \rangle$$

  where $L_{t-1} \triangleq \sum_{s=1}^{t-1} \boldsymbol{\ell}_s \in \mathbb{R}^N$ is the cumulative loss vector.

  ⇨ The solution is actually a one-hot vector, which is very ***unstable***.

A Natural Solution for PEA

- Follow the Leader (FTL)

  Idea: Select the expert that performs best so far, namely,

$$\boldsymbol{p}_t^{\text{FTL}} = \arg\min_{\boldsymbol{p}\in\Delta_N} \langle \boldsymbol{p}, L_{t-1} \rangle$$

where $L_{t-1} \triangleq \sum_{s=1}^{t-1} \boldsymbol{\ell}_s \in \mathbb{R}^N$ is the cumulative loss vector.

⇨ The solution is actually a one-hot vector, which is very ***unstable***.

⇨ Replacing the 'max' operation in FTL by ***'softmax'***.

Hedge: Algorithm

Hedge Algorithm

At each round $t = 1, 2, \cdots$

(1) compute $\boldsymbol{p}_t \in \Delta_N$ such that $\boldsymbol{p}_t(i) \propto \exp\left(-\eta L_{t-1}(i)\right)$ for $i \in [N]$

(2) the player submits $\boldsymbol{p}_t$, suffers loss $\langle \boldsymbol{p}_t, \boldsymbol{\ell}_t \rangle$, and observes loss $\boldsymbol{\ell}_t \in \mathbb{R}^N$

(3) update $L_t = L_{t-1} + \boldsymbol{\ell}_t$

FTL update

$$\boldsymbol{p}_t = \arg\max_{\boldsymbol{p} \in \Delta_N} \langle \boldsymbol{p}, -L_{t-1} \rangle$$

Hedge update

$$\boldsymbol{p}_t(i) \propto \exp\left(-\eta L_{t-1}(i)\right), \forall i \in [N]$$

Hedge: Regret Bound

Theorem 1. *Suppose that* $\forall t \in [T]$ *and* $i \in [N], 0 \leq \boldsymbol{\ell}_t(i) \leq 1$, *then Hedge with learning rate* η *guarantees*

$$\mathrm{Regret}_T \leq \frac{\ln N}{\eta} + \eta T,$$

which is of order $\mathcal{O}(\sqrt{T \ln N})$ *if* η *is optimally set as* $\sqrt{(\ln N)/T}$.

Proof. We present a 'potential-based' proof here, where the potential is defined as

$$\Phi_t \triangleq \frac{1}{\eta} \ln \left(\sum_{i=1}^{N} \exp\left(-\eta L_t(i)\right) \right).$$

Proof of Hedge Regret Bound

Proof.

$$\Phi_t - \Phi_{t-1} = \frac{1}{\eta}\ln\left(\frac{\sum_{i=1}^N \exp\left(-\eta L_t(i)\right)}{\sum_{i=1}^N \exp\left(-\eta L_{t-1}(i)\right)}\right) \qquad \Phi_t \triangleq \frac{1}{\eta}\ln\left(\sum_{i=1}^N \exp\left(-\eta L_t(i)\right)\right)$$

$$= \frac{1}{\eta}\ln\left(\sum_{i=1}^N\left(\frac{\exp\left(-\eta L_{t-1}(i)\right)}{\sum_{i=1}^N \exp\left(-\eta L_{t-1}(i)\right)}\exp(-\eta \boldsymbol{\ell}_t(i))\right)\right)$$

$$= \frac{1}{\eta}\ln\left(\sum_{i=1}^N \boldsymbol{p}_t(i)\exp\left(-\eta\boldsymbol{\ell}_t(i)\right)\right) \qquad \text{(update step of } \boldsymbol{p}_t\text{)}$$

$$\le \frac{1}{\eta}\ln\left(\sum_{i=1}^N \boldsymbol{p}_t(i)\left(1-\eta\boldsymbol{\ell}_t(i)+\eta^2\boldsymbol{\ell}_t^2(i)\right)\right) \qquad (\forall x\ge 0, e^{-x}\le 1-x+x^2)$$

$$= \frac{1}{\eta}\ln\left(1-\eta\left\langle \boldsymbol{p}_t,\boldsymbol{\ell}_t\right\rangle+\eta^2\sum_{i=1}^N \boldsymbol{p}_t(i)\boldsymbol{\ell}_t^2(i)\right)$$

Proof of Hedge Regret Bound

Proof.

$$\Phi_t - \Phi_{t-1} = \frac{1}{\eta} \ln \left(1 - \eta \langle \boldsymbol{p}_t, \boldsymbol{\ell}_t \rangle + \eta^2 \sum_{i=1}^{N} \boldsymbol{p}_t(i) \boldsymbol{\ell}_t^2(i) \right)$$

$$\leq - \langle \boldsymbol{p}_t, \boldsymbol{\ell}_t \rangle + \eta \sum_{i=1}^{N} \boldsymbol{p}_t(i) \boldsymbol{\ell}_t^2(i) \qquad (\ln(1+x) \leq x)$$

Summing over t, we have

$$\sum_{t=1}^{T} \langle \boldsymbol{p}_t, \boldsymbol{\ell}_t \rangle \leq \Phi_0 - \Phi_T + \eta \sum_{t=1}^{T} \sum_{i=1}^{N} \boldsymbol{p}_t(i) \boldsymbol{\ell}_t^2(i) \qquad \Phi_t \triangleq \frac{1}{\eta} \ln \left(\sum_{i=1}^{N} \exp\left(-\eta L_t(i)\right) \right)$$

$$\leq \frac{\ln N}{\eta} - \frac{1}{\eta} \ln \left(\exp\left(-\eta L_T(i^\star)\right) \right) + \eta \sum_{t=1}^{T} \sum_{i=1}^{N} \boldsymbol{p}_t(i) \boldsymbol{\ell}_t^2(i)$$

$$\leq \frac{\ln N}{\eta} + L_T(i^\star) + \eta \sum_{t=1}^{T} \sum_{i=1}^{N} \boldsymbol{p}_t(i) \boldsymbol{\ell}_t^2(i)$$

Proof of Hedge Regret Bound

Proof.

$$\sum_{t=1}^{T} \langle \boldsymbol{p}_t, \boldsymbol{\ell}_t \rangle \leq \frac{\ln N}{\eta} + L_T(i^\star) + \eta \sum_{t=1}^{T} \sum_{i=1}^{N} \boldsymbol{p}_t(i) \boldsymbol{\ell}_t^2(i)$$

Rearranging the term gives

$$\begin{aligned}\sum_{t=1}^{T} \langle \boldsymbol{p}_t, \boldsymbol{\ell}_t \rangle - L_T(i^\star) &\leq \frac{\ln N}{\eta} + \eta \sum_{t=1}^{T} \sum_{i=1}^{N} \boldsymbol{p}_t(i) \boldsymbol{\ell}_t^2(i) \\ &\leq \frac{\ln N}{\eta} + \eta T \qquad (\boldsymbol{\ell}_t(i) \leq 1)\end{aligned}$$

Thus, setting $\eta = \sqrt{\ln N / T}$ yields

$$\text{Regret}_T \leq \frac{\ln N}{\eta} + \eta T = 2\sqrt{T \ln N}. \qquad \square$$

Lower bound of PEA

- As above, we have proved the regret bound for Hedge:

$$\mathrm{Regret}_T \leq 2\sqrt{T \ln N}$$

- A natural question: can we further improve the bound?

Theorem 2 (Lower Bound of PEA). *For any algorithm $\mathcal{A}$, we have that*

$$\sup_{T,N} \max_{\ell_1,\ldots,\ell_T} \frac{\mathrm{Regret}_T}{\sqrt{T \ln N}} \geq \frac{1}{\sqrt{2}}.$$

Hedge achieves minimax optimal regret (up to a constant of $2\sqrt{2}$) for PEA.

Lower bound of PEA

Theorem 2 (Lower Bound of PEA). *For any algorithm $\mathcal{A}$, we have that*

$$\sup_{T,N} \max_{\boldsymbol{\ell}_1,\ldots,\boldsymbol{\ell}_T} \frac{\mathrm{Regret}_T}{\sqrt{T \ln N}} \geq \frac{1}{\sqrt{2}}.$$

Proof. We construct the 'hard' instance by randomization. Let $\mathcal{D}$ be the uniform distribution over $\{0, 1\}$. We have

$$\max_{\boldsymbol{\ell}_1,\ldots,\boldsymbol{\ell}_T} \mathrm{Regret}_T \geq \mathbb{E}_{\boldsymbol{\ell}_1,\ldots,\boldsymbol{\ell}_T \overset{\text{i.i.d.}}{\sim} \mathcal{D}^N} [\mathrm{Regret}_T]$$

(conditional expectation decomposition)

$$= \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{\ell}_1,\ldots,\boldsymbol{\ell}_{t-1}} \mathbb{E}_{\boldsymbol{\ell}_t} [\langle \boldsymbol{p}_t, \boldsymbol{\ell}_t \rangle \mid \boldsymbol{\ell}_{t-1}, \ldots, \boldsymbol{\ell}_1] - \mathbb{E}_{\boldsymbol{\ell}_1,\ldots,\boldsymbol{\ell}_T} \left[\min_{i \in [N]} \sum_{t=1}^{T} \boldsymbol{\ell}_t(i) \right]$$

$$= \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{\ell}_1,\ldots,\boldsymbol{\ell}_{t-1}} \langle \boldsymbol{p}_t, \mathbb{E}_{\boldsymbol{\ell}_t} [\boldsymbol{\ell}_t \mid \boldsymbol{\ell}_{t-1}, \ldots, \boldsymbol{\ell}_1] \rangle - \mathbb{E}_{\boldsymbol{\ell}_1,\ldots,\boldsymbol{\ell}_T} \left[\min_{i \in [N]} \sum_{t=1}^{T} \boldsymbol{\ell}_t(i) \right]$$

Lower bound of PEA

Theorem 2 (Lower Bound of PEA). *For any algorithm $\mathcal{A}$, we have that*

$$\sup_{T,N} \max_{\boldsymbol{\ell}_1,\ldots,\boldsymbol{\ell}_T} \frac{\text{Regret}_T}{\sqrt{T \ln N}} \geq \frac{1}{\sqrt{2}}.$$

Proof.

$$\max_{\boldsymbol{\ell}_1,\ldots,\boldsymbol{\ell}_T} \text{Regret}_T \geq \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{\ell}_1,\ldots,\boldsymbol{\ell}_{t-1}} \langle \boldsymbol{p}_t, \mathbb{E}_{\boldsymbol{\ell}_t}[\boldsymbol{\ell}_t \mid \boldsymbol{\ell}_{t-1},\ldots,\boldsymbol{\ell}_1] \rangle - \mathbb{E}_{\boldsymbol{\ell}_1,\ldots,\boldsymbol{\ell}_T}\left[\min_{i\in[N]} \sum_{t=1}^{T} \boldsymbol{\ell}_t(i)\right]$$

$$= T/2 - \mathbb{E}_{\boldsymbol{\ell}_1,\ldots,\boldsymbol{\ell}_T}\left[\min_{i\in[N]} \sum_{t=1}^{T} \boldsymbol{\ell}_t(i)\right] = \mathbb{E}_{\boldsymbol{\ell}_1,\ldots,\boldsymbol{\ell}_T}\left[\max_{i\in[N]} \sum_{t=1}^{T} \left(\frac{1}{2} - \boldsymbol{\ell}_t(i)\right)\right]$$

($\boldsymbol{\ell}_t \overset{\text{i.i.d.}}{\sim} \mathcal{D}$ with $\mathcal{D}$ be the uniform distribution over $\{0,1\}$)

$$= \frac{1}{2}\mathbb{E}_{\sigma_1,\ldots,\sigma_T}\left[\max_{i\in[N]} \sum_{t=1}^{T} \sigma_t(i)\right],$$

(σ_t for $i \in [N], t \in [T]$ are i.i.d. Rademacher random variables)

Lower bound of PEA

Theorem 2 (Lower Bound of PEA). *For any algorithm $\mathcal{A}$, we have that*

$$\sup_{T,N} \max_{\boldsymbol{\ell}_1,\ldots,\boldsymbol{\ell}_T} \frac{\text{Regret}_T}{\sqrt{T \ln N}} \geq \frac{1}{\sqrt{2}}.$$

Proof.

$$\max_{\boldsymbol{\ell}_1,\ldots,\boldsymbol{\ell}_T} \text{Regret}_T \geq \frac{1}{2}\mathbb{E}_{\sigma_1,\ldots,\sigma_T}\left[\max_{i\in[N]} \sum_{t=1}^{T} \sigma_t(i)\right]$$

($\sigma_t(i)$ for $i \in [N], t \in [T]$ are i.i.d. Rademacher random variables)

Using the result from probability theory (*Prediction, Learning, and Games*, Chapter 3.7) of Rademacher variables,

$$\Longrightarrow \quad \lim_{T\to\infty} \lim_{N\to\infty} \frac{\mathbb{E}_{\sigma_1,\ldots,\sigma_T}\left[\max_{i\in[N]} \sum_{t=1}^{T} \sigma_t(i)\right]}{\sqrt{T \ln N}} = \sqrt{2}. \qquad \square$$

Upper Bound and Lower Bound

Theorem 1. *Suppose that* $\forall t \in [T]$ *and* $i \in [N], 0 \le \boldsymbol{\ell}_t(i) \le 1$, *then Hedge with learning rate* η *guarantees*

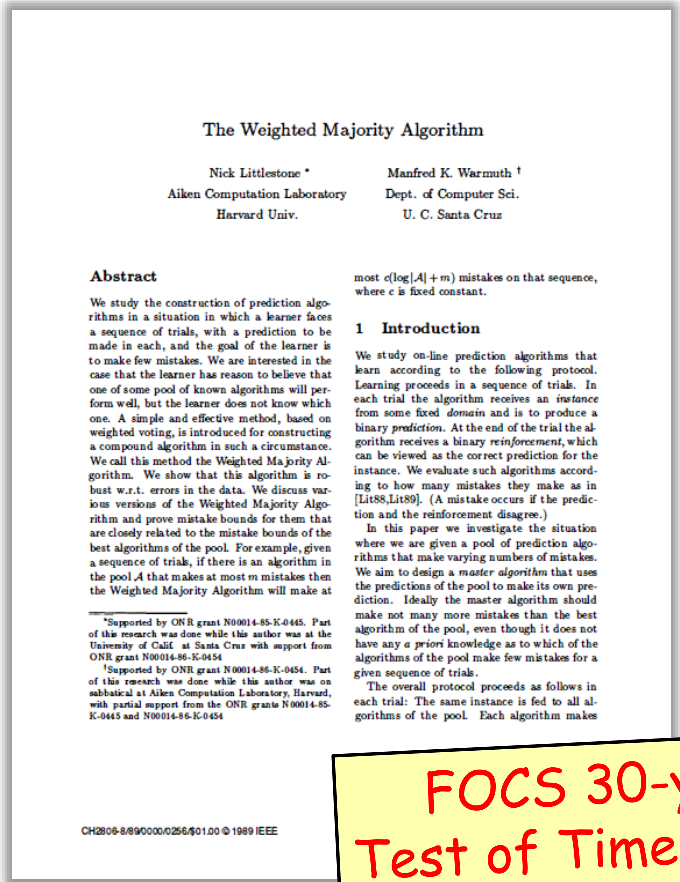
$$\text{Regret}_T \le \frac{\ln N}{\eta} + \eta T,$$

which is of order $\mathcal{O}(\sqrt{T \ln N})$ *if* η *is optimally set as* $\sqrt{(\ln N)/T}$.

Theorem 2 (Lower Bound of PEA)**.** *For any algorithm* $\mathcal{A}$, *we have that*

$$\sup_{T,N} \max_{\boldsymbol{\ell}_1,\ldots,\boldsymbol{\ell}_T} \frac{\text{Regret}_T}{\sqrt{T \ln N}} \ge \frac{1}{\sqrt{2}}.$$

Prediction with Expert Advice: history bits



The Weighted Majority Algorithm

Nick Littlestone *
Aiken Computation Laboratory
Harvard Univ.

Manfred K. Warmuth †
Dept. of Computer Sci.
U. C. Santa Cruz

Abstract

We study the construction of prediction algorithms in a situation in which a learner faces a sequence of trials, with a prediction to be made in each, and the goal of the learner is to make few mistakes. We are interested in the case that the learner has reason to believe that one of some pool of known algorithms will perform well, but the learner does not know which one. A simple and effective method, based on weighted voting, is introduced for constructing a compound algorithm in such a circumstance. We call this method the Weighted Majority Algorithm. We show that this algorithm is robust w.r.t. errors in the data. We discuss various versions of the Weighted Majority Algorithm and prove mistake bounds for them that are closely related to the mistake bounds of the best algorithms of the pool. For example, given a sequence of trials, if there is an algorithm in the pool $\mathcal{A}$ that makes at most m mistakes then the Weighted Majority Algorithm will make at most $c(\log|\mathcal{A}| + m)$ mistakes on that sequence, where c is fixed constant.

1 Introduction

We study on-line prediction algorithms that learn according to the following protocol. Learning proceeds in a sequence of trials. In each trial the algorithm receives an *instance* from some fixed *domain* and is to produce a binary *prediction*. At the end of the trial the algorithm receives a binary *reinforcement*, which can be viewed as the correct prediction for the instance. We evaluate such algorithms according to how many mistakes they make as in [Lit88,Lit89]. (A mistake occurs if the prediction and the reinforcement disagree.)

In this paper we investigate the situation where we are given a pool of prediction algorithms that make varying numbers of mistakes. We aim to design a *master algorithm* that uses the predictions of the pool to make its own prediction. Ideally the master algorithm should make not many more mistakes than the best algorithm of the pool, even though it does not have any *a priori* knowledge as to which of the algorithms of the pool make few mistakes for a given sequence of trials.

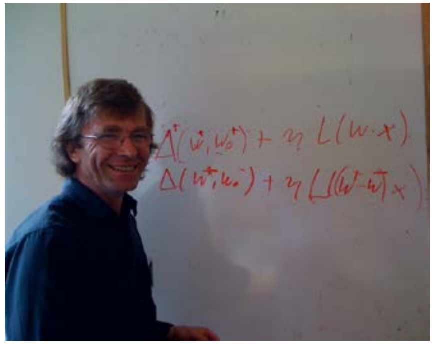
The overall protocol proceeds as follows in each trial: The same instance is fed to all algorithms of the pool. Each algorithm makes

*Supported by ONR grant N00014-85-K-0445. Part of this research was done while this author was at the University of Calif. at Santa Cruz with support from ONR grant N00014-86-K-0454

†Supported by ONR grant N00014-86-K-0454. Part of this research was done while this author was on sabbatical at Aiken Computation Laboratory, Harvard, with partial support from the ONR grants N00014-85-K-0445 and N00014-86-K-0454

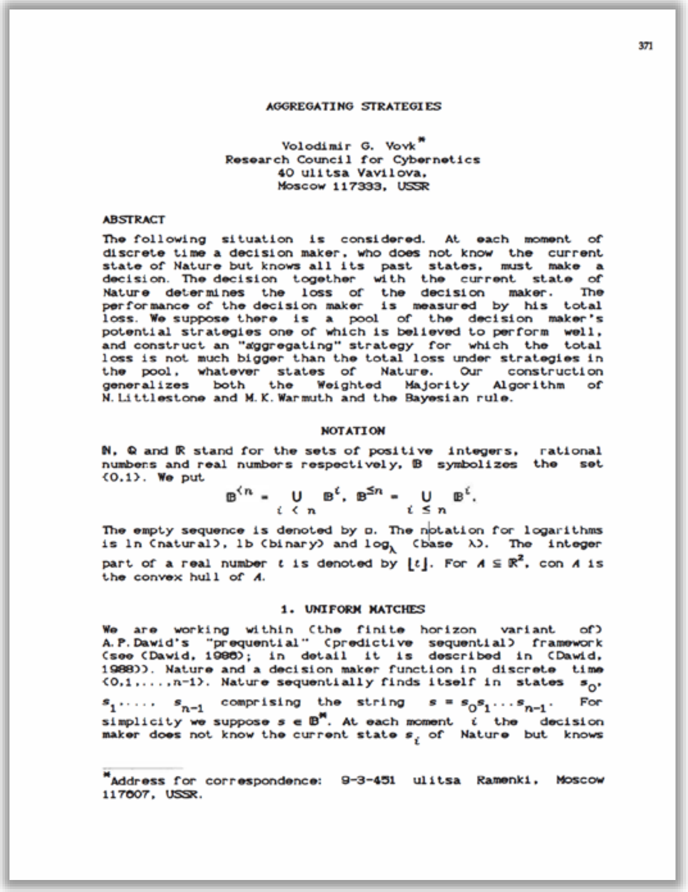
CH2806-8/89/0000/0256$01.00 © 1989 IEEE

FOCS 30-year Test of Time Award!



Manfred Warmuth

UC Santa Cruz



371

AGGREGATING STRATEGIES

Volodimir G. Vovk*
Research Council for Cybernetics
40 ulitsa Vavilova,
Moscow 117333, USSR

ABSTRACT

The following situation is considered. At each moment of discrete time a decision maker, who does not know the current state of Nature but knows all its past states, must make a decision. The decision together with the current state of Nature determines the loss of the decision maker. The performance of the decision maker is measured by his total loss. We suppose there is a pool of the decision maker's potential strategies one of which is believed to perform well, and construct an "aggregating" strategy for which the total loss is not much bigger than the total loss under strategies in the pool, whatever states of Nature. Our construction generalizes both the Weighted Majority Algorithm of N. Littlestone and M. K. Warmuth and the Bayesian rule.

NOTATION

$\mathbb{N}$, $\mathbb{Q}$ and $\mathbb{R}$ stand for the sets of positive integers, rational numbers and real numbers respectively, $\mathbb{B}$ symbolizes the set {0,1}. We put

$$\mathbb{B}^{<n} = \bigcup_{i<n} \mathbb{B}^i, \quad \mathbb{B}^{\le n} = \bigcup_{i \le n} \mathbb{B}^i.$$

The empty sequence is denoted by □. The notation for logarithms is ln (natural), lb (binary) and $\log_\lambda$ (base λ). The integer part of a real number t is denoted by $\lfloor t \rfloor$. For $A \subseteq \mathbb{R}^2$, con A is the convex hull of A.

1. UNIFORM MATCHES

We are working within (the finite horizon variant of) A. P. Dawid's "prequential" (predictive sequential) framework (see (Dawid, 1986); in detail it is described in (Dawid, 1988)). Nature and a decision maker function in discrete time {0,1,...,n-1}. Nature sequentially finds itself in states s_0, s_1, ..., s_{n-1} comprising the string $s = s_0 s_1 \ldots s_{n-1}$. For simplicity we suppose $s \in \mathbb{B}^n$. At each moment i the decision maker does not know the current state s_i of Nature but knows

*Address for correspondence: 9-3-451 ulitsa Ramenki, Moscow 117607, USSR.

Volodimir G. Vovk
Royal Holloway,
University of London

Nick Littlestone and Manfred K. Warmuth.
"The Weighted Majority Algorithm." FOCS 1989: 256-261.

Volodimir G. Vovk. "Aggregating Strategies." COLT 1990: 371-383.

PEA vs. OCO

Prediction with Expert Advice

At each round $t = 1, 2, \cdots$

(1) the player first picks a weight $\boldsymbol{p}_t$ from a simplex Δ_N;

(2) and simultaneously environments pick an loss vector $\boldsymbol{\ell}_t \in \mathbb{R}^N$;

(3) the player suffers loss $f_t(\boldsymbol{p}_t) \triangleq \langle \boldsymbol{p}_t, \boldsymbol{\ell}_t \rangle$, observes $\boldsymbol{\ell}_t$ and updates the model.

require domain to be a simplex $\mathcal{X} = \Delta_N$ ⇧ linear loss $f_t(\mathbf{x}) \triangleq \langle \mathbf{x}, \boldsymbol{\ell}_t \rangle$

PEA is a ***special case*** of OCO!

Online Convex Optimization

At each round $t = 1, 2, \cdots$

(1) the player first picks a model $\mathbf{x}_t \in \mathcal{X}$;

(2) and simultaneously environments pick an online function $f_t : \mathcal{X} \to \mathbb{R}$;

(3) the player suffers loss $f_t(\mathbf{x}_t)$, observes f_t and updates the model.

Deploying OGD to PEA

- PEA is a special case of OCO:

Why not directly deploy OGD (proposed in last lecture) to address PEA?

> **Theorem 4** (Regret bound for OGD). *Under Assumption 1, 2 and 3, online gradient descent (OGD) with step sizes* $\eta_t = \frac{D}{G\sqrt{t}}$ *for* $t \in [T]$ *guarantees:*
>
> $$\text{Regret}_T = \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x} \in \mathcal{X}} \sum_{t=1}^{T} f_t(\mathbf{x}) \le \frac{3}{2} GD\sqrt{T}.$$

Regret guarantee: $D = \max_{\mathbf{x},\mathbf{y} \in \Delta_N} \|\mathbf{x} - \mathbf{y}\|_2 = \sqrt{2}$ $\quad G = \max_{\boldsymbol{\ell}_t \in \mathbb{R}^N} \|\boldsymbol{\ell}_t\|_2 = \sqrt{N}$

$$\Longrightarrow \text{Regret}_T = \sum_{t=1}^{T} \langle \boldsymbol{p}_t, \boldsymbol{\ell}_t \rangle - \min_{\boldsymbol{p} \in \Delta_N} \sum_{t=1}^{T} \langle \boldsymbol{p}, \boldsymbol{\ell}_t \rangle \le \mathcal{O}(\sqrt{TN})$$

Deploying OGD to PEA

- OGD for PEA Problem:

$$D = \max_{\mathbf{x},\mathbf{y}\in\Delta_N} \|\mathbf{x}-\mathbf{y}\|_2 = \sqrt{2} \qquad G = \max_{\boldsymbol{\ell}_t\in\mathbb{R}^N} \|\boldsymbol{\ell}_t\|_2 = \sqrt{N}$$

$$\Rightarrow \quad \text{Regret}_T = \sum_{t=1}^{T} \langle \boldsymbol{p}_t, \boldsymbol{\ell}_t \rangle - \min_{\boldsymbol{p}\in\Delta_N} \sum_{t=1}^{T} \langle \boldsymbol{p}, \boldsymbol{\ell}_t \rangle \leq \mathcal{O}(\sqrt{TN})$$

- A natural question: is the $\mathcal{O}(\sqrt{TN})$ regret bound tight enough?
  - recall that the lower bound of PEA is $\Omega(\sqrt{T \ln N})$
  - OGD is not optimal with respect to N (number of experts)

Deploying OGD to PEA

- PEA is a special case of OCO:

Why not directly deploy OGD (proposed in last lecture) to address PEA?

Theorem 4 (Regret bound for OGD). *Under Assumption 1, 2 and 3, online gradient descent (OGD) with step sizes $\eta_t = \frac{D}{G\sqrt{t}}$ for $t \in [T]$ guarantees:*

$$\text{Regret}_T = \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x}\in\mathcal{X}} \sum_{t=1}^{T} f_t(\mathbf{x}) \leq \frac{3}{2} GD\sqrt{T}.$$

Regret guarantee: $D = \max_{\mathbf{x},\mathbf{y}\in\Delta_N} \|\mathbf{x}-\mathbf{y}\|_2 = \sqrt{2}$ $\quad G = \max_{\boldsymbol{\ell}_t\in\mathbb{R}^N} \|\boldsymbol{\ell}_t\|_2 = \sqrt{N}$

$$\Longrightarrow \text{Regret}_T = \sum_{t=1}^{T} \langle \boldsymbol{p}_t, \boldsymbol{\ell}_t \rangle - \min_{\boldsymbol{p}\in\Delta_N} \sum_{t=1}^{T} \langle \boldsymbol{p}, \boldsymbol{\ell}_t \rangle \leq \mathcal{O}(\sqrt{TN})$$

Why OGD Fails for PEA

- PEA has a special structure whereas general OCO doesn't have.

Convex Problem	*PEA Problem*
Domain: convex set $\mathcal{X}$	Domain: simplex $\mathcal{X} = \Delta_N$
Online function: convex function f_t	Online function: linear $f_t(\boldsymbol{p}) \triangleq \langle \boldsymbol{p}, \boldsymbol{\ell}_t \rangle$
Lower Bound: $\Omega(GD\sqrt{T})$	Lower Bound: $\Omega(\sqrt{T \ln N})$

Why OGD Fails for PEA

- Remember that for the general OCO, we linearized the function to analyze the first gradient descent lemma:

$$
\begin{aligned}
\|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2 &= \|\Pi_{\mathcal{X}}[\mathbf{x}_t - \eta_t \nabla f(\mathbf{x}_t)] - \mathbf{x}^\star\|^2 \quad \text{(GD)} \\
&\le \|\mathbf{x}_t - \eta_t \nabla f(\mathbf{x}_t) - \mathbf{x}^\star\|^2 \quad \text{(Pythagoras Theorem)} \\
&= \|\mathbf{x}_t - \mathbf{x}^\star\|^2 - 2\eta_t \langle \nabla f(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}^\star \rangle + \eta_t^2 \|\nabla f(\mathbf{x}_t)\|^2 \\
&\le \|\mathbf{x}_t - \mathbf{x}^\star\|^2 - 2\eta_t (f(\mathbf{x}_t) - f^\star) + \eta_t^2 \|\nabla f(\mathbf{x}_t)\|^2
\end{aligned}
$$

(convexity: $f(\mathbf{x}_t) - f^\star = f(\mathbf{x}_t) - f(\mathbf{x}^\star) \le \langle \nabla f(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}^\star \rangle$)

- So, linearized loss is not the essence, but the ***simplex domain*** of the PEA problem is worthy specifically considering.

Why OGD Fails for PEA

- Recall that for general OCO, we update the model as follows:

General Online Convex Optimization

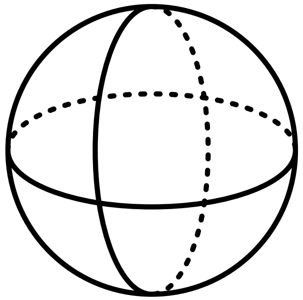
OGD:

$$\mathbf{x}_{t+1} = \Pi_{\mathcal{X}}\left[\mathbf{x}_t - \eta_t \nabla f(\mathbf{x}_t)\right]$$

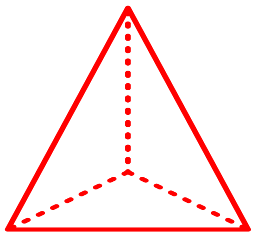
Proximal type update:

$$\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}\in\mathcal{X}} \langle \mathbf{x}, \eta_t \nabla f_t(\mathbf{x}_t)\rangle + \frac{1}{2}\|\mathbf{x}-\mathbf{x}_t\|_2^2$$

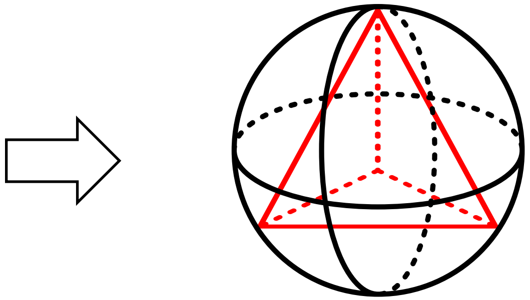
- In PEA, is it proper to use 2-norm (ball) to measure distance?



Ball



Simplex



A ball is too pessimistic (loose) to measure a simplex!

Why OGD Fails for PEA

- Recall that for general OCO, we update the model as follows:

General Online Convex Optimization

OGD:

$$\mathbf{x}_{t+1} = \Pi_{\mathcal{X}} \left[\mathbf{x}_t - \eta_t \nabla f(\mathbf{x}_t) \right]$$

Proximal type update:

$$\mathbf{x}_{t+1} = \arg\min_{\mathbf{x} \in \mathcal{X}} \langle \mathbf{x}, \eta_t \nabla f_t(\mathbf{x}_t) \rangle + \frac{1}{2} \|\mathbf{x} - \mathbf{x}_t\|_2^2$$

- In PEA, is it proper to use 2-norm (ball) to measure distance?

⇨ We need to find an ***alternative distance measure*** for the ***special structure*** in PEA.

Reinvent Hedge Algorithm

⇨ We need to find an ***alternative distance measure*** for the ***special structure*** in PEA.

- Intuitively, for Euclidean space, 2-norm is the most natural measure:

$$\|\mathbf{x} - \mathbf{y}\|_2^2$$

- For PEA problem
  - the decision can be viewed as a distribution within the simplex
  - for two distributions P and Q, KL divergence is a natural measure:

$$\mathrm{KL}(P\|Q) = \sum_{x \in \mathcal{X}} P(x) \log\left(\frac{P(x)}{Q(x)}\right)$$

Reinvent Hedge Algorithm

Theorem 3. *Consider $f_t(\boldsymbol{p}) = \langle \boldsymbol{p}, \boldsymbol{\ell}_t \rangle$. An online learning algorithm that updates the model following*

$$\boldsymbol{p}_{t+1} = \arg\min_{\boldsymbol{p} \in \Delta_N} \eta \langle \boldsymbol{p}, \nabla f_t(\boldsymbol{p}_t) \rangle + \mathrm{KL}(\boldsymbol{p} \| \boldsymbol{p}_t)$$

is equal to Hedge update, i.e.,

$$\boldsymbol{p}_{t+1}(i) \propto \exp\left(-\eta L_t(i)\right) \text{ for all } i \in [N].$$

Proof.

$$\begin{aligned} \boldsymbol{p}_{t+1} &= \arg\min_{\boldsymbol{p} \in \Delta_N} \eta \langle \boldsymbol{p}, \nabla f_t(\boldsymbol{p}_t) \rangle + \mathrm{KL}(\boldsymbol{p} \| \boldsymbol{p}_t) \\ &= \arg\min_{\boldsymbol{p} \in \Delta_N} \underbrace{\eta \langle \boldsymbol{p}, \nabla f_t(\boldsymbol{p}_t) \rangle - \sum_{i=1}^{N} \boldsymbol{p}(i) \ln\left(\frac{\boldsymbol{p}_t(i)}{\boldsymbol{p}(i)}\right)}_{F(\boldsymbol{p})} \qquad \text{(definition of KL divergence)} \end{aligned}$$

Proof

Proof. $$\boldsymbol{p}_{t+1} = \underset{\boldsymbol{p}\in\Delta_N}{\arg\min}\; \underbrace{\eta\langle \boldsymbol{p}, \nabla f_t(\boldsymbol{p}_t)\rangle - \sum_{i=1}^{N} \boldsymbol{p}(i)\ln\left(\frac{\boldsymbol{p}_t(i)}{\boldsymbol{p}(i)}\right)}_{F(\boldsymbol{p})}$$ (definition of KL divergence)

$F(\boldsymbol{p})$ is convex, therefore we minimize $\boldsymbol{p}$ by taking $\nabla_{\boldsymbol{p}} F(\boldsymbol{p}) = 0$:

$$\forall i \in [N], \quad \eta(\nabla f_t(\boldsymbol{p}_t))_i - \ln(\boldsymbol{p}_t(i)) - 1 + \ln(\boldsymbol{p}_{t+1}(i)) = 0$$

$$\Rightarrow \quad \begin{aligned} \boldsymbol{p}_{t+1}(i) &= \exp\big(-\eta\nabla(f_t(\boldsymbol{p}_t))_i + \ln(\boldsymbol{p}_t(i)) + 1\big) \\ &= \boldsymbol{p}_t(i)\exp\big(-\eta\boldsymbol{\ell}_t(i) + 1\big) \qquad (f_t(\boldsymbol{p}) = \langle \boldsymbol{p}, \boldsymbol{\ell}_t\rangle) \\ &= \boldsymbol{p}_{t-1}(i)\exp\big(-\eta(\boldsymbol{\ell}_t(i) + \boldsymbol{\ell}_{t-1}(i)) + 2\big) \\ &= \dots \\ &= \boldsymbol{p}_0(i)\exp\big(-\eta L_t(i) + t\big) \end{aligned}$$

$\boldsymbol{p}_{t+1}(i) \propto \exp(-\eta L_t(i))$ for all $i \in [N]$

Reinvent Hedge Algorithm

- Proximal update rule for OGD:

$$\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}\in\mathcal{X}} \eta_t \langle \mathbf{x}, \nabla f_t(\mathbf{x}_t)\rangle + \frac{1}{2}\|\mathbf{x}-\mathbf{x}_t\|_2^2$$

- Proximal update rule for Hedge:

$$\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}\in\mathcal{X}} \eta_t \langle \mathbf{x}, \nabla f_t(\mathbf{x}_t)\rangle + \mathrm{KL}(\mathbf{x}\|\mathbf{x}_t)$$

- More possibility: changing the distance measure to a more general form using ***Bregman divergence***

$$\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}\in\mathcal{X}} \eta_t \langle \mathbf{x}, \nabla f_t(\mathbf{x}_t)\rangle + \mathcal{D}_\psi(\mathbf{x},\mathbf{x}_t)$$