Expert Learning

Giacomo Boracchi, Francesco Trovò

May 4th, 2022

Politecnico di Milano, DEIB

francesco1.trovo@polimi.it

Short Bio

- PhD @DEIB in 2015
- Working @AirLab since 2015
- Assistant Professor @DEIB since 2020

Scientific interests:

- Multi-armed bandit algorithms
- Internet economic scenarios
- Health scenarios

Lecture Overview

Online Learning

Binary prediction Space

Expert Learning

- ***Continuous action space***
- ***Discrete action space***

Infinite Number of Experts

Online Learning

Model and Regret

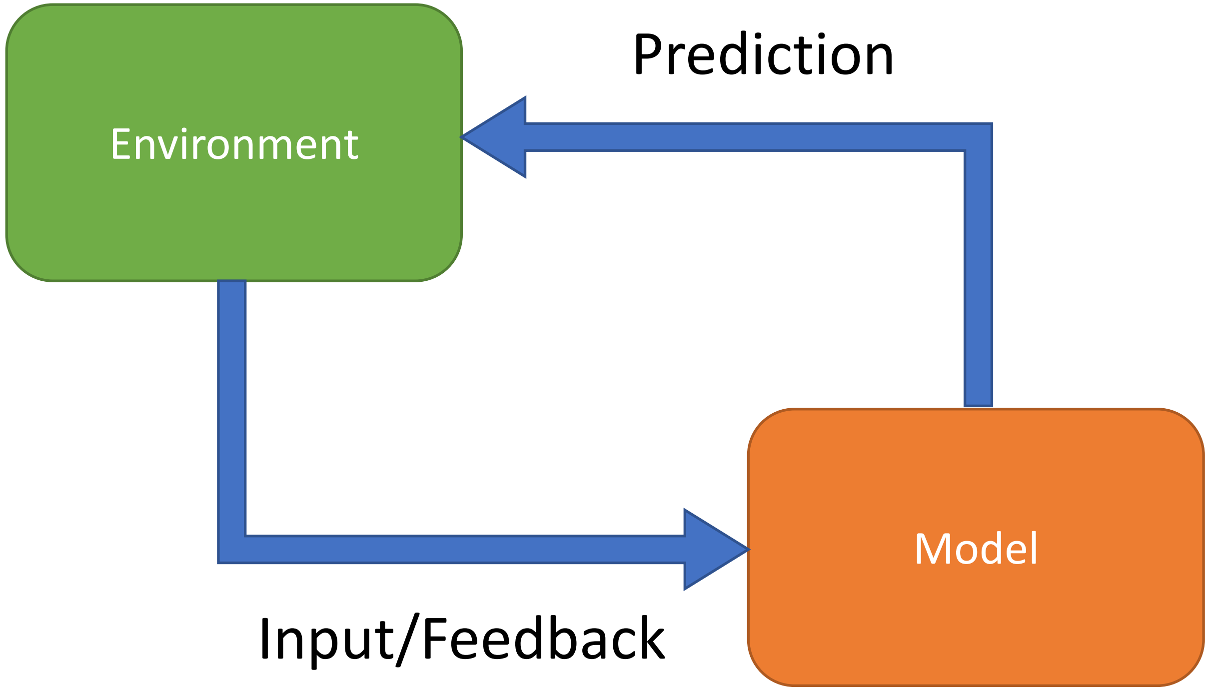
Boracchi, Trovò

General Framework

The environment is changing or adversarial

Required to handle streaming data

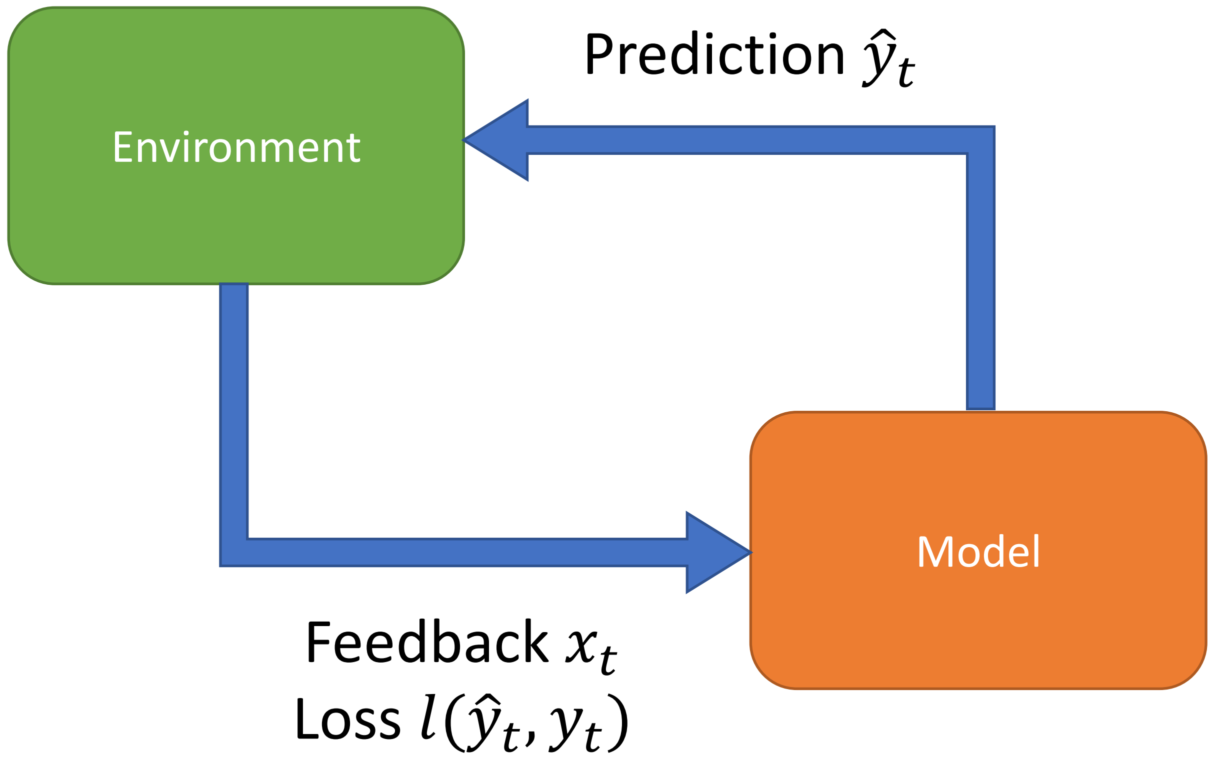
- We need to learn
- We need to adapt



Online Learning Framework

at each round t

- we generate a prediction $\hat{y}_t$
- the environment chooses y_t
- we suffer a loss of $l(\hat{y}_t, y_t)$
- we might get feedback x_t
- we update the model we use for prediction



Examples

- What will be the *rain precipitation* next month?
- What will be the *price of this stock* tomorrow?

- How *many iPad* will be sold during the next quarter?
- How many *contacts will have this webpage* in the next hour?

Commonly the prediction also corresponds to an action or decision to be performed at a specific time

Weather Prediction

Website providing weather forecasts for tomorrow:

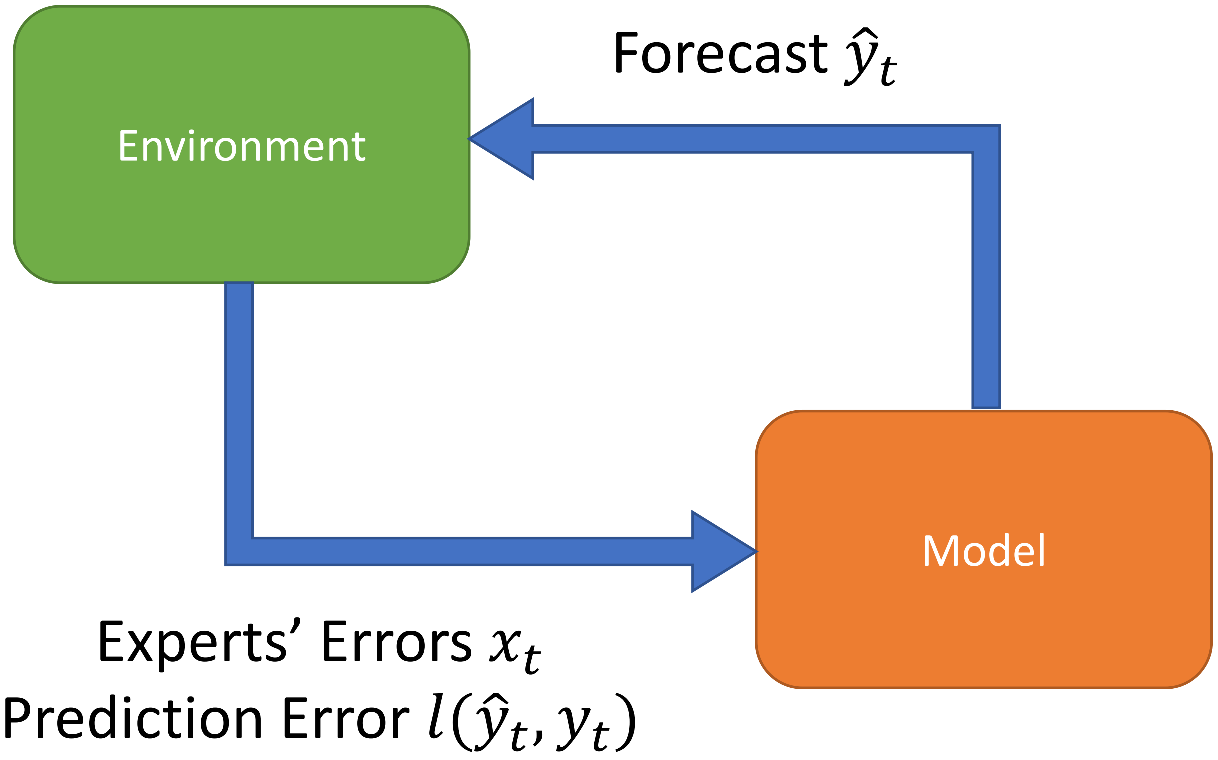
- We are no expert in meteorology
- We can look the to other forecasting services and choose among them

We can look at the results of all the forecasting services a posteriori of the selection

Objective:

We would like to be as good as the best expert

Weather Prediction as an Online Learning Problem



Pricing Problem

- You have a new product
- You do not know the optimal price (price providing the largest revenue)
    - Ask for a market study
    - Rely on historical information (e.g., NERDs salary)
    - Try to learn the price without losing too much money

We have a set of options $D \ = \ \{1\$, 10\$ \ 50\$ \ 100\$ \ 500\$ \ 799\$ \ 2000\$\}$

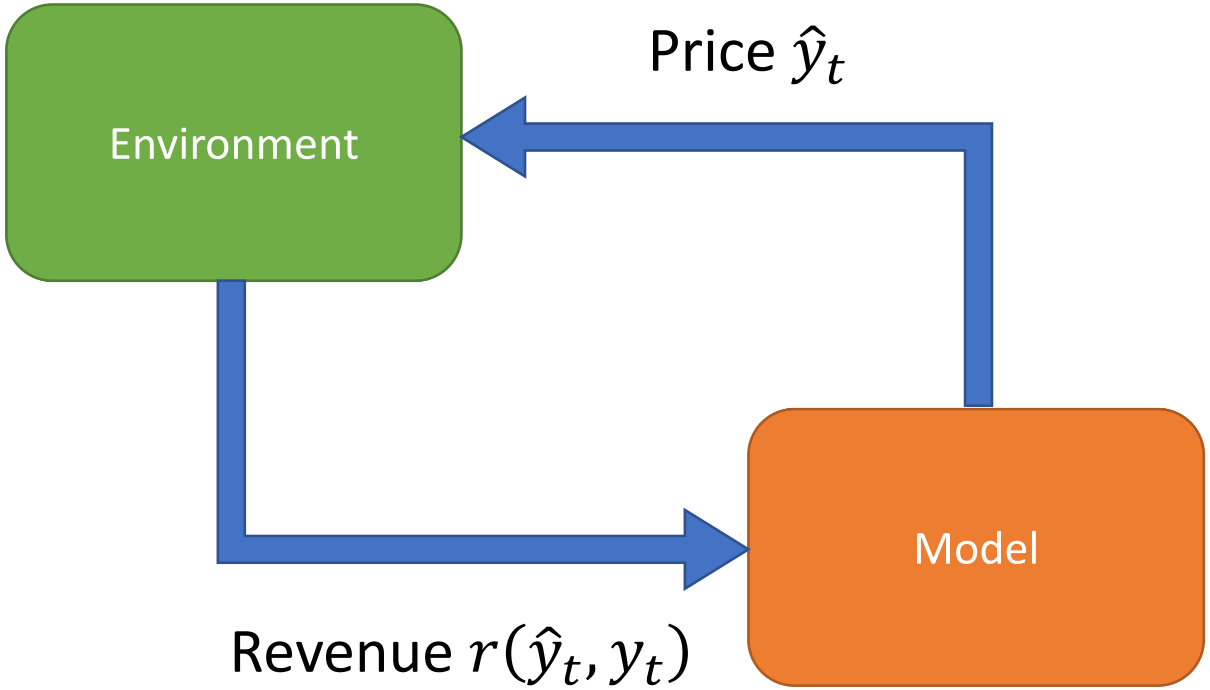
maximize the reward per round $\hat{y} \cdot \mu$

Each time you select a suboptimal price you lose some money

Each customer will provide you with a feedback about a single price

Pricing as an Online Learning Problem

$$r(\hat{y}_t, y_t) = 1 - l(\hat{y}_t, y_t)$$



No information on other prices x_t

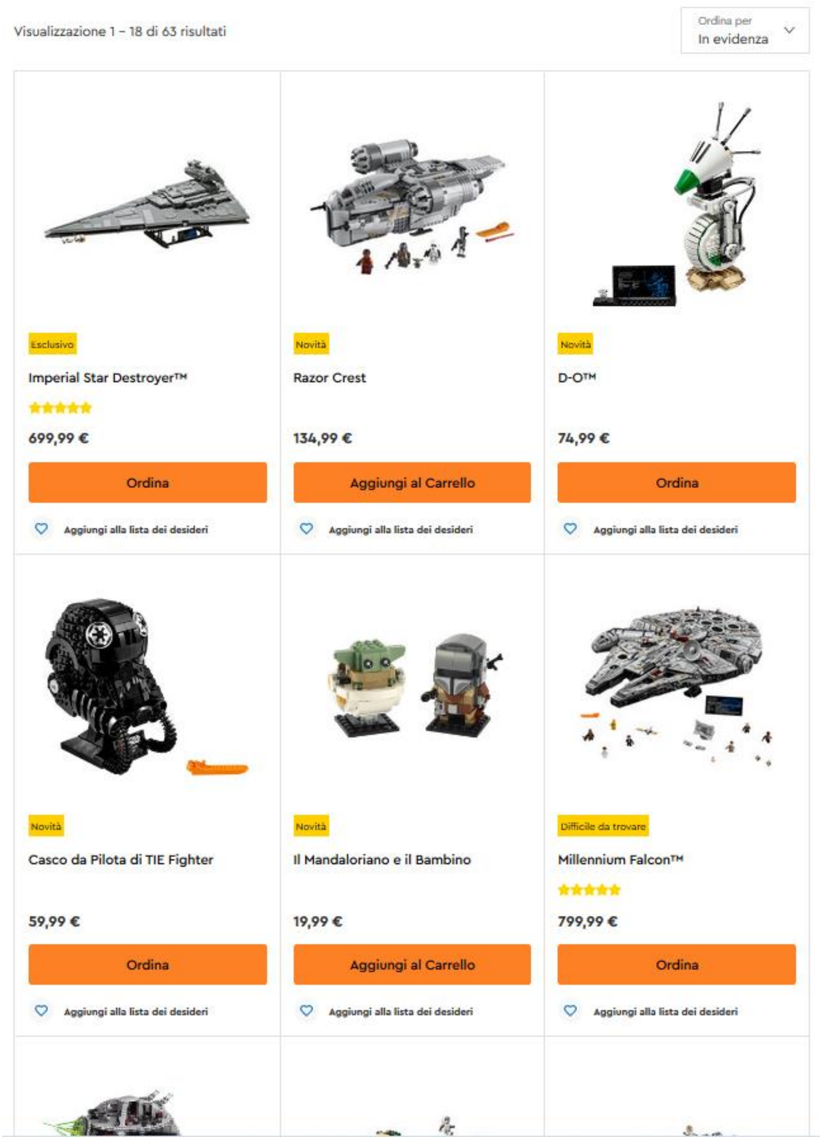
More Complex Pricing Problem

You have a catalog of products

You want to set a price for each one of them

You do not want to waste time in estimating the price for each one product

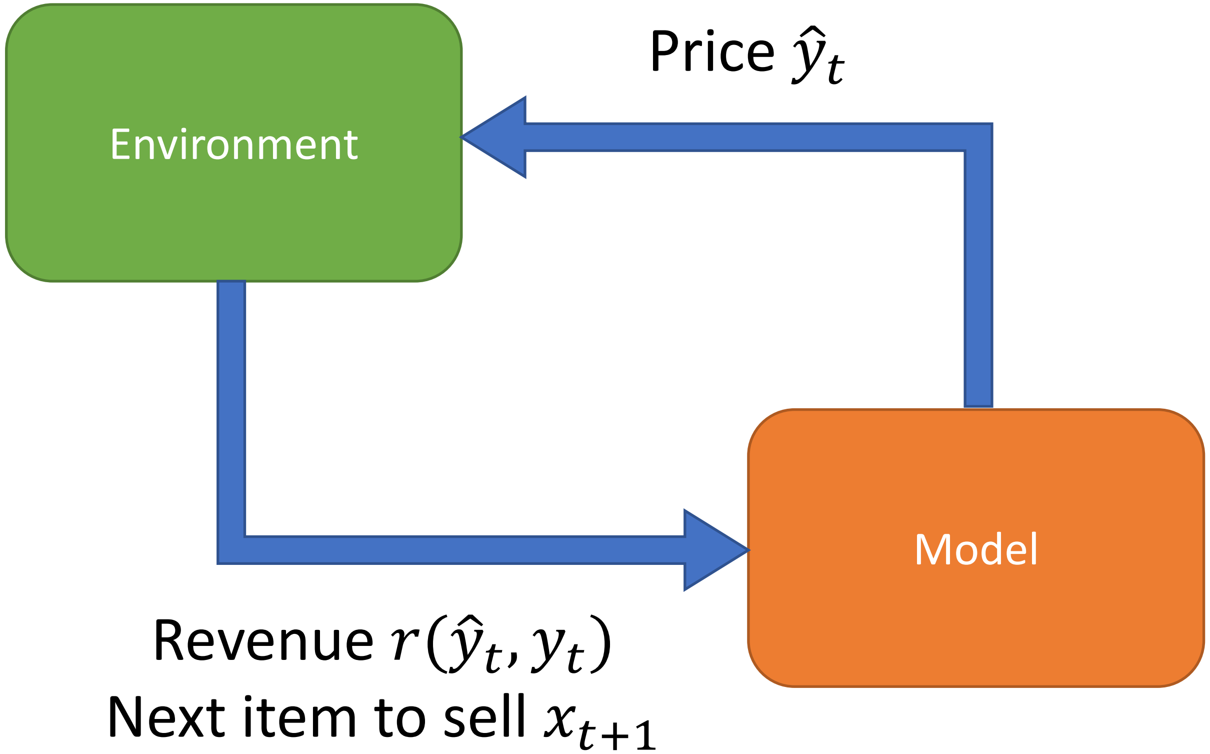
You need to learn a more general rule determining the price given the product characteristics



Boracchi, Trovò

Pricing as an Online Learning Problem

$$r(\hat{y}_t, y_t) = 1 - l(\hat{y}_t, y_t)$$



Online Learning Problems in Your Research

Do you think that some of the problems in your research area can be modeled as Online Learning ones?

Keep this question in mind during the entire course length

Online Learning vs. Classical Machine Learning

- We cannot ensure that real processes are fully stochastic
- We cannot measure expected performances
- Data are coming in a sequence (stream)
- The **training and testing phases** are rarely separated in real-world problems
- Massive datasets are usually provided as a stream
- We have some spatial and computational constraints

Online Learning vs. Learning in NSE

- Online Learning does not require a statistical characterization of the process
- Provides strong theoretical results vs. practical approach
- Starting with no information on the system vs. requires an initial model

This requires to study of simple models and, only after that, their extension to more complex scenarios

Online Learning vs. Reinforcement Learning

Online is more like a meta-approach

Some RL algorithms are Online Learning algorithms too (e.g., Q-learning)

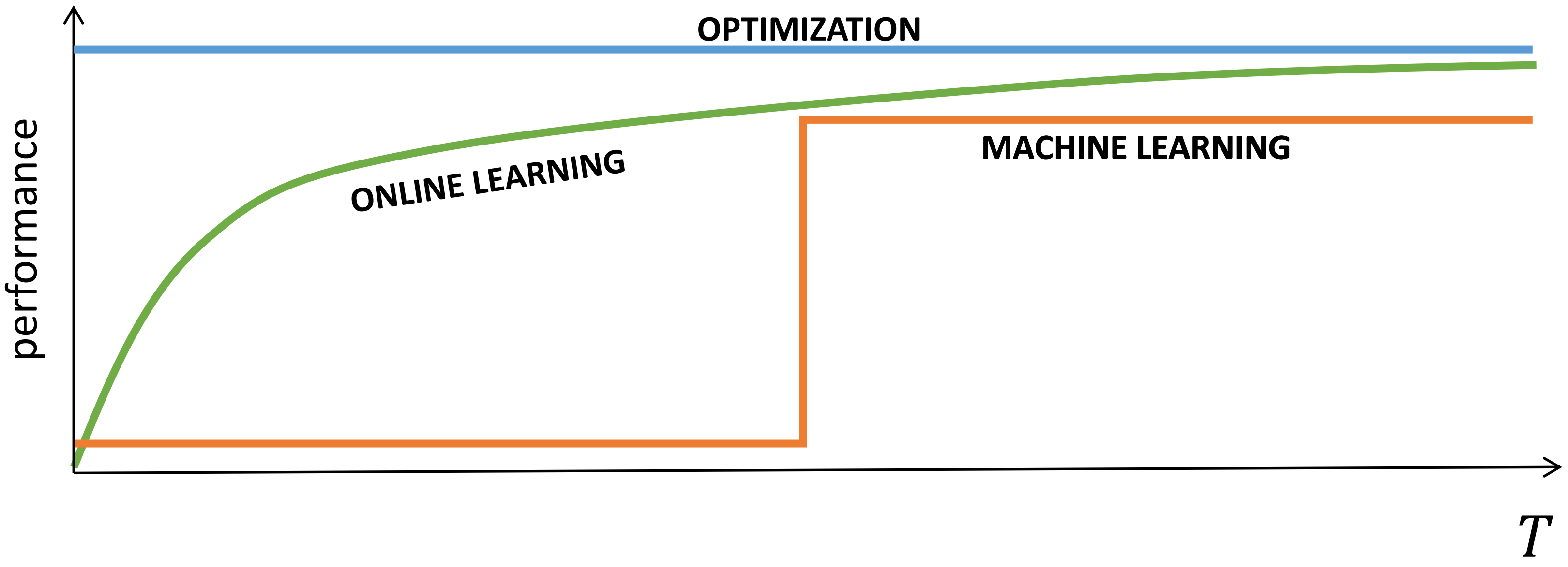
Some Online Learning algorithms have been developed for specific RL scenarios (e.g., UCB1)

RL usually has some statistical assumptions on the reward (or loss) and on the evolution of the process

Online Learning also handles data generated from an opponent (game theoretical approach)

Algorithms Evaluation

We cannot use concepts like estimation error, accuracy, precision, and recall



Regret $R_T(A)$: loss of the designed algorithm w.r.t. a clairvoyant (optimal) choice among the ones in a given set

Regret Definition

We want to compare our algorithm with a baseline:

Definition: Regret

Given an algorithm A, selecting a prediction $\hat{y}_t$ at round t, and a clairvoyant algorithm A^*, selecting a prediction y_t^* at round t, the Regret of A over a time horizon of T rounds is:

$$R_T(A) = \sum_{t=1}^{T} [l(\hat{y}_t, y_t) - l(y_t^*, y_t)]$$

The definition of the clairvoyant algorithm might change depending on the setting:

- Best prediction $y_t^* = \min_{y \in C} \sum_{t=1}^{T} l(y, y_t)$
- Best constant average prediction $y_t^* = \min_{y} \sum_{t=1}^{T} E[l(y, y_t)]$

No-Regret Algorithms

We are interested in algorithms which provides a regret which, asymptotically, is sublinear in the time horizon T

Definition:

An algorithm A is said to be no-regret if:

$$\lim_{T \to +\infty} \frac{R_T(A)}{T} = 0$$

This way we are assured to have a vanishing regret as the time horizon progresses

As a byproduct we are also converging to the optimal solution

Different Online Learning Problems

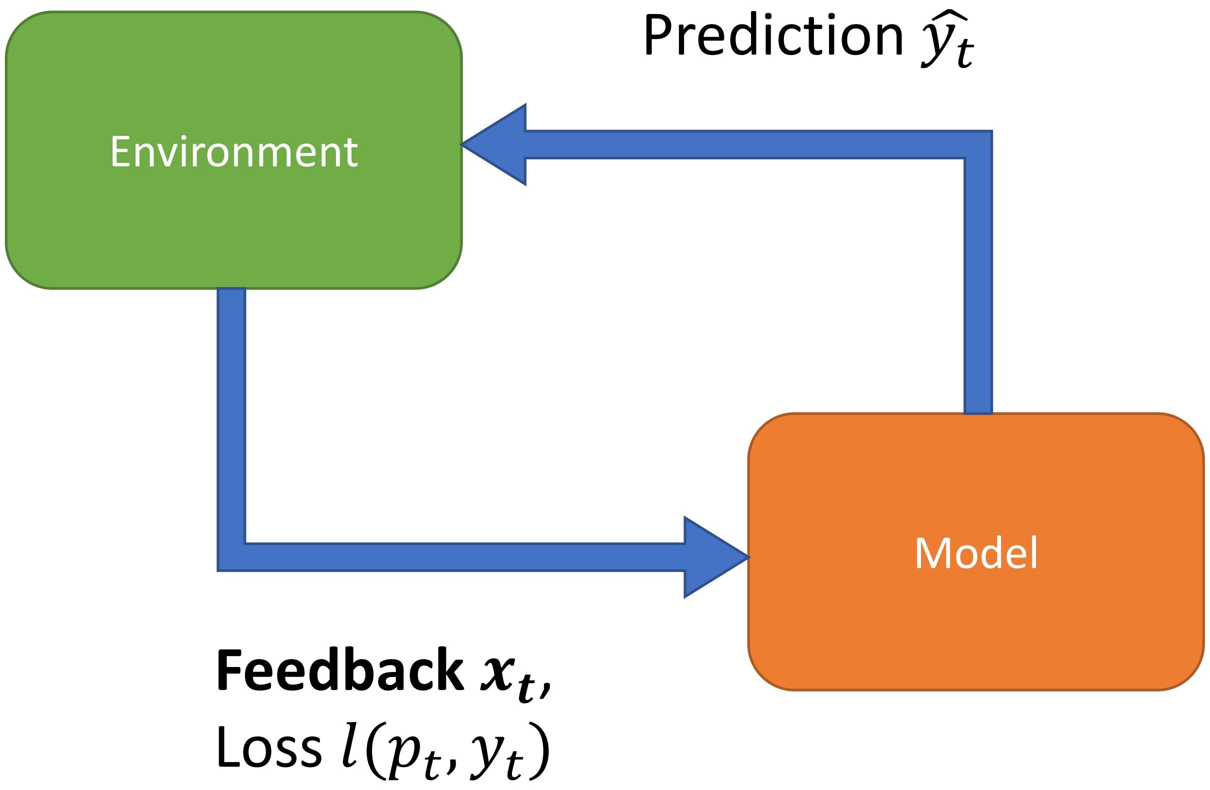
- **Expert learning**

Loss for all the possible choices
$x_t = \{l(p, y_t), \forall p \in D\}$

- Multi-Armed Bandit (MAB)

No feedback $x_t = ()$

- Partial Monitoring
- …



The Prediction Game

Choose the following elements:

- the outcome space Y
- the decision space D
- the performance function $l(\hat{y}, y)$

At each round t

the environment chooses $y_t \in Y$ and the learner chooses $\hat{y}_t \in D$

the learner suffers a loss $l(\hat{y}_t, y_t)$

the environment reveals $\boldsymbol{y_t}$

Binary Prediction Space

Example: Heads or Tails?

A game in which we can choose heads or tails, but we do not know if the coin is fair and if changes over time

We can ask an audience for advice

Among the audience, we have a hidden superhero with the power of predicting the future (only 30 sec ahead), but we do not know who he/she is

Binary Prediction

Simple case: we want to predict a string of bits:

- the outcome space $Y = \{0, 1\}$
- the decision space $D = \{0, 1\}$
- the performance function $l(\hat{y}, y) = 1\{\hat{y} \neq y\}$

No assumption on the distribution in the outcome space and its variation over time

We rely on the information provided by N experts

Each expert generates a prediction $f_{i,t} \in D$ $(i \in \{1, \ldots N\})$

Single Perfect Expert

Assume to have one of the experts which predicts perfectly the sequence:

$$\exists i, \forall t, l(f_{i,t}, y_t) = 0$$

Halving algorithm:

initialize the experts' weights $w_{i,t} = 1$

at each round t

we collect the experts' predictions $f_{i,t}$ for the experts with $w_{i,t} = 1$

predict $\hat{y}_t = 1$ if the majority of them predicts so, $\hat{y}_t = 0$, otherwise

observe y_t

set $w_{i,t} = 0$ for each expert s.t. $f_{i,t} \neq y_t$

Analysis of the Halving Algorithm

Theorem

The Halving algorithm applied to a binary prediction problem with N experts makes at most $m \leq \lfloor \log_2 N \rfloor$ mistakes if at least one of the expert is perfect

Proof:

Define $W_t = \sum_i w_{i,t}$

At $t = 0$ we have $m = 0$ and $W_0 = N$

At each mistake we have $W_m \leq \frac{W_{m-1}}{2}$

Recursively we have $W_m \leq \frac{W_0}{2^m}$

Since at least one expert is perfect we have $W_m \geq 1$

Finally, $\frac{N}{2^m} \geq 1 \Rightarrow m \leq \lfloor \log_2 N \rfloor$

Imperfect Experts

If no expert is perfect, we want to relate the number of mistakes made m with the ones made by the best expert m^*

We cannon set a weight to zero for a single error $\rightarrow$ we shrink it by a factor β

Weighted Halving algorithm:

- initialize the experts' weights $w_{i,t} = 1$
- at each round t
  - we collect the experts' predictions $f_{i,t}$
  - predict $\hat{y}_t$ according to the weighted majority
  - observe y_t
  - set $w_{i,t} \leftarrow \beta w_{i,t}$ for each expert s.t. $f_{i,t} \neq y_t$

Analysis of the Weighted Halving Algorithm

Theorem

The Weighted Halving algorithm applied to a binary prediction problem with N experts and shrinking factor $\beta < 1$ makes at most $m \leq \left\lfloor \frac{\log_2 N - m^* \log_2 \beta}{1 - \log_2 (1+\beta)} \right\rfloor$ mistakes if at least one of the expert makes at most m^* mistakes

Proof:

At $t = 0$ we have $m = 0$ and $W_0 \; = \; N$

At each mistake we have $W_m \leq \frac{W_{m-1}}{2} + \beta \frac{W_{m-1}}{2} = (1 + \beta) \frac{W_{m-1}}{2}$

Recursively we have $W_m \leq (1 + \beta)^m \frac{W_0}{2^m}$

Since at least one expert made at most m^* mistakes we have $W_m \geq \beta^{m^*}$

Finally, $\frac{N(1+\beta)^m}{2^m} \geq \beta^{m^*} \Rightarrow m \leq \left\lfloor \frac{\log_2 N - m^* \log_2 \beta}{1 - \log_2 (1+\beta)} \right\rfloor$

Weather Prediction

Website providing weather forecasts for tomorrow:

- We are no expert in meteorology
- We can look the to other forecasting services and choose among them

We can look at the results of all the forecasting services a posteriori of the selection

Objective:

We would like to be as good as the best expert

Expert Learning

Convex Loss

Weather Prediction++

Website providing **rain mm** for tomorrow:

- We are no expert in meteorology
- We can look the to other forecasting services and choose among them

We can look at the results of all the forecasting services a posteriori of the selection



Continuous Prediction Space

- the outcome space Y is arbitrary
- the decision space D is a convex subset of $\mathbb{R}^s$
- the performance function $l(\hat{y}, y)$
  - is bounded, for simplicity $l(\hat{y}, y) \in [0,1]$
  - convex in the first argument $l(\cdot, y)$ for each $y \in Y$

In this context, we cannot count the number of mistakes. Instead, we use:

Definition: Regret

$$R_n(A) := \sum_{t=1}^{n} l(\hat{y}_t, y_t) - \min_{i \in \{1,\ldots,N\}} \sum_{t=1}^{n} l(f_{i,t}, y_t)$$

Concept of Experts

An expert can be anything:

- Fixed over time $f_{i,t} = c_i$
- Adaptive experts $f_{i,t} = f_i(x)$, where x is a context
- Learning experts $f_{i,t} = f_i(t, y_1, \dots, y_{t-1})$
- Experts can even form a coalition against the learner

We are not aware about how the expert is able to provide its prediction, nor to replicate the forecast process

Exponentially Weighted Average Forecaster

Exponentially Weighted Average (EWA)

initialize the experts' weights $w_{i,t} = 1$

at each round t

we collect the experts' predictions $f_{i,t}$

predict $\hat{y}_t = \frac{\sum_{i=1}^{N} w_{i,t-1} f_{i,t}}{\sum_{i=1}^{N} w_{i,t-1}}$

observe y_t and suffer loss $l(\hat{y}_t, y_t)$

update the weights $w_{i,t} \leftarrow w_{i,t-1} \exp(-\eta\, l(f_{i,t}, y_t))$ for each expert

The more an expert suffer loss, the less is used to provide a prediction

We need only to store the normalized weight $\widehat{w}_{i,t} = \frac{w_{i,t}}{\sum_{i=1}^{N} w_{i,t}}$ for each expert

EWA Regret Bound

Theorem

The EWA algorithm applied to a continuous prediction problem with N experts and with parameter η has regret:

$$R_n(EWA) \leq \frac{logN}{\eta} + \frac{\eta n}{8}$$

We use the following lemma:

Theorem: Hoeffding Inequality

Let X be a random variable with $a \ \leq \ X \ \leq \ b$. Then for any $s \ \in \ \mathbb{R}$:

$$\log(E[\exp(sX)]) \leq s\ E[X] + \frac{s^2(b-a)\text{^}2}{8}$$

EWA Regret Bound

Proof: Let us analyze the quantity $W_t \; = \sum_{i=1}^{N} w_{i,t}$

Step 1:

$$log \frac{W_{n+1}}{W_1} = \log\left(\sum_{i=1}^{N} w_{i,n+1}\right) - logN \geq \log\left(\max_i w_{i,n+1}\right) - logN$$

$$= -\eta \min_i \sum_{t=1}^{n} l\left(f_{i,t}, y_t\right) \; - log\; N$$

Step 2:

$$log \frac{W_{t+1}}{W_t} = \log\left(\sum_{i=1}^{N} \frac{w_{i,t}}{W_t} \; \exp(-\eta l\left(f_{i,t}, y_t\right))\right) = \log\left(E[exp(-\eta l\left(f_{i,t}, y_t\right)]\right)$$

$$\leq -\eta E\left[l\left(f_{i,t}, y_t\right)\right] + \frac{\eta^2}{8} \leq -\eta\left[l\left(E\left[f_{i,t}\right], y_t\right)\right] + \frac{\eta^2}{8} \leq -\eta l(\hat{y}_t, y_t) + \frac{\eta^2}{8}$$

EWA Regret Bound

Step 3:

$$\log \frac{W_{n+1}}{W_1} = \sum_{t=1}^{n} \log \frac{W_{t+1}}{W_t}$$

$$-\eta \min_i \sum_{t=1}^{n} l\left(f_{i,t}, y_t\right) - log\ N \leq \log \frac{W_{n+1}}{W_1} = \sum_{t=1}^{n} \log \frac{W_{t+1}}{W_t} \leq \sum_{t=1}^{n} \left(-\eta l(\hat{y}_t, y_t) + \frac{\eta^2}{8}\right)$$

$$-\eta \min_i \sum_{t=1}^{n} l\left(f_{i,t}, y_t\right) - log\ N \leq -\eta \sum_{t=1}^{n} l(\hat{y}_t, y_t) + \frac{n\eta^2}{8}$$

$$\sum_{t=1}^{n} l(\hat{y}_t, y_t) - \min_i \sum_{t=1}^{n} l\left(f_{i,t}, y_t\right) \leq \frac{logN}{\eta} + \frac{n\eta}{8}$$

Parameter Tuning

$$w_{i,t} \leftarrow w_{i,t-1} \exp(-\eta \, l(f_{i,t}, y_t))$$

We need to find a way to set η:

- large values for η: we converge to a single expert which might be the wrong one
- small values for η: we converge to the correct expert, but it takes a long time

This is reflected in the regret bound too:

$$R_n(EWA) \leq \frac{logN}{\eta} + \frac{\eta n}{8}$$

We can minimize the bound in terms of time horizon n and number of expert N choosing $\eta = \sqrt{\frac{8 \, logN}{n}}$ getting:

$$R_n(EWA) \leq \sqrt{\frac{n \, logN}{2}}$$

Parameter Tuning

If we do not know the time horizon:

Theorem

The EWA algorithm applied to a continuous prediction problem with N experts and with parameter $\eta_t = \sqrt{\frac{8 \log N}{t}}$ has regret:

$$R_n(EWA) \leq \sqrt{\frac{n \log N}{2}} + \sqrt{\frac{logN}{8}}$$

Proof: nontrivial extension of the proof with η constant (see Cesa-Bianchi et al. 2006)

Scientific Question about the Result

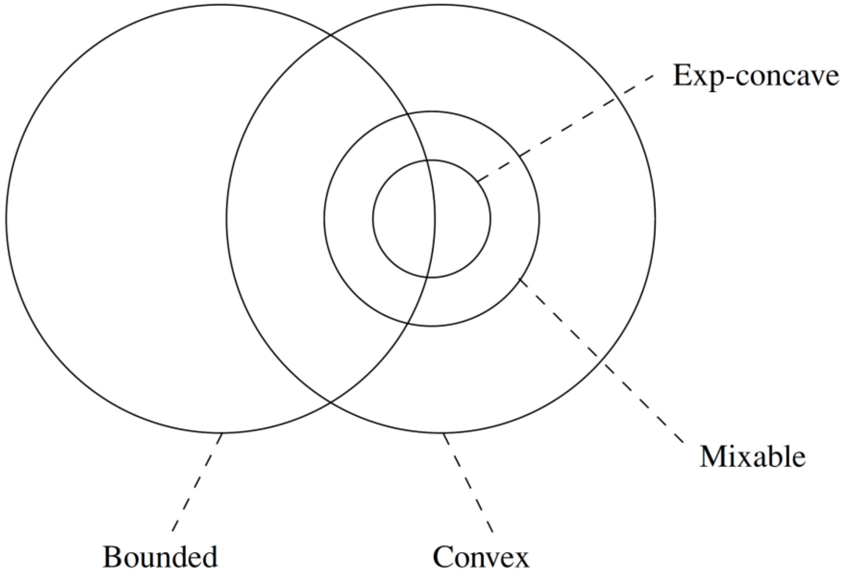
Is this a proper result for an Online Learning algorithm?

Is this the best algorithm we might design for this specific setting?

Are there other algorithms that provide better results in more specific cases?

Are there algorithms providing better performance in practice?

Lower Bound



Bounded and convex: $\Theta(n\ logN)$, matched by the EWA forecaster

Mixable: $c\ logN$, not necessarily matched by the EWA

Exp concave: $c\ logN$, matched by the EWA

Quadratic Loss

Let us restrict on a specific loss function: the quadratic one

$$l(\hat{y}, y) = (\hat{y} - y)^2$$

In this case a simple strategy has also strong theoretical guarantees:

Follow the Leader (FL)
at each round t
collect the experts' predictions $f_{i,t}$
predict $\hat{y}_t = f_{E,t}$ where $E = \arg\min_{i \in \{1,\ldots,N\}} \sum_{s=1}^{t-1} l(f_{i,s}, y_s)$
observe y_t and suffer loss $l(\hat{y}_t, y_t)$

FL Bound

Theorem

The FL algorithm applied to a continuous prediction problem against constant experts and quadratic loss has regret:

$$R_n(FL) \leq 8(\log n + 1)$$

Proof (sketch):

In this specific case, the FL algorithm chooses $\widehat{y_t} = \sum_{s=1}^{t-1} y_s$

Step 1: show that the FL forecaster knowing also the losses suffered at round t performs as well as the best constant expert

Step 2: show that the prediction of the original FL differs from the previous one for a factor of at most $\epsilon_t \leq \frac{8}{t}$

Step 3: then the regret is bounded by $\sum_{s=1}^{t} \epsilon_s \leq 8(\log n + 1)$

Other Convex Optimization Algorithms

- Gradient-based exponentially weighted average forecaster: instead of the loss in the weight update we use the gradient of the loss
- Multilinear forecaster: weights are updated as

$$w_{i,t+1} \leftarrow w_{i,t} \left(1 + \eta\, h\big(f_{i,t}, y_t\big)\right)$$

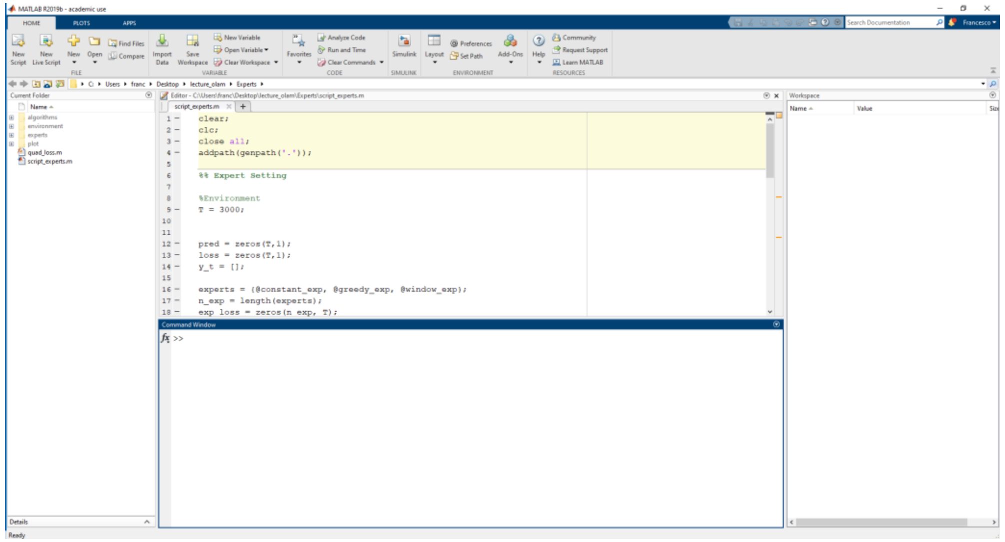
where $h(\cdot,\cdot)$ is a payoff function

- Greedy forecaster: chooses the expert by minimizing the worst-case loss at the next step combined with its loss up to now
- Online Gradient Descent: at each round update the prediction using a single step in the direction of the gradient

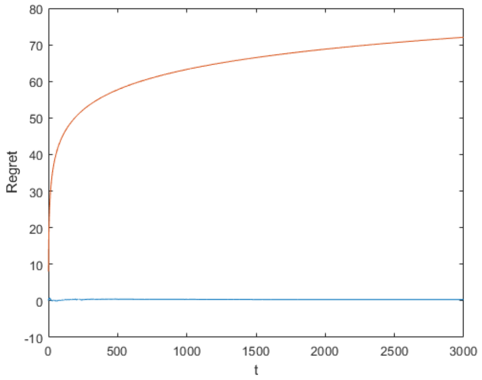
Matlab Exercise

Given an Expert environment:

- Implement the EWA forecaster
- Implement the FL forecaster



Draw the regret for both algorithms and compare them with the FL bound



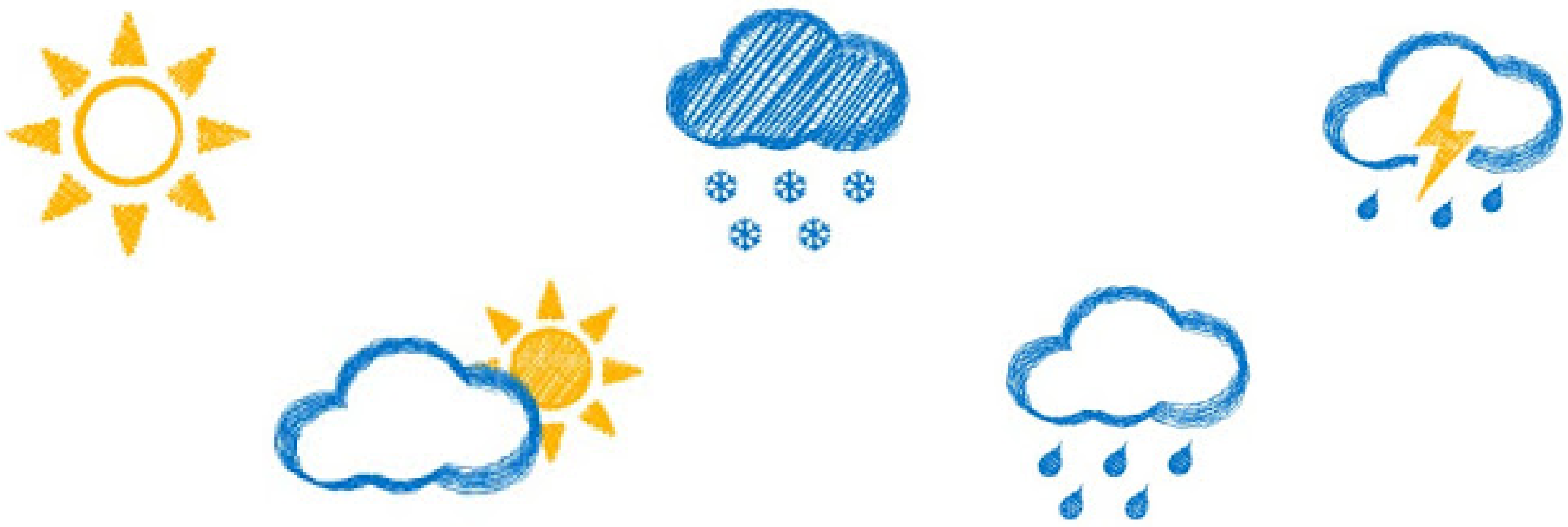
Expert Learning

Non-Convex Loss

Weather Prediction#

Website providing **forecasting** for tomorrow:

- We are no expert in meteorology
- We can look the to other forecasting services and choose among them



Online Discrete Prediction

- the outcome space Y is discrete ($|Y| > 2$)
- the decision space $D = Y$
- the performance function $l(\hat{y}, y) = 1(\hat{y} \neq y)$

We count mistakes w.r.t. a class of N experts providing **constant** prediction $f_1, \dots, f_N$

Definition: Regret

$$R_n(A) \coloneqq \sum_{t=1}^{n} l(\hat{y}_t, y_t) - \min_{i \in \{1,\dots,N\}} \sum_{t=1}^{n} l(f_i, y_t)$$

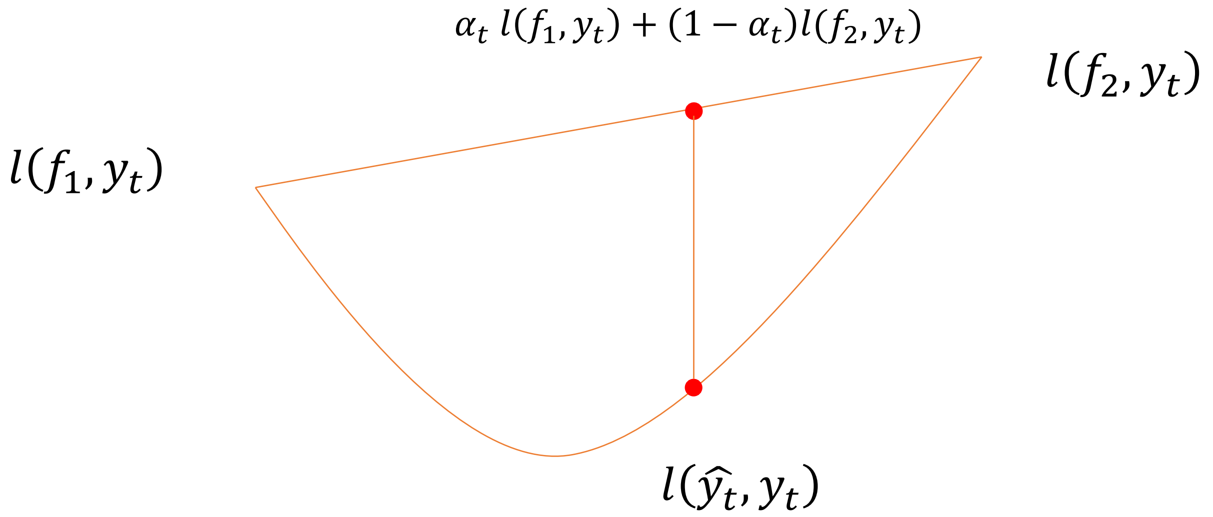
....it should be as difficult as the continuous case...

Need for Convex Loss: Idea

Assume to have only two experts

The prediction we are providing is a convex combination of the twos:

$$\hat{y}_t = \alpha_t\, f_1 + (1 - \alpha_t) f_2$$



The convex combination of the experts' losses overestimates the real loss

Deterministic Algorithms

The loss function is not convex in the first argument

Regret counterexample: two classes and the experts are $f_1 = 0$ and $f_2 = 1$

For any algorithm A there exists at least a sequence $y_1(A), \dots y_n(A)$ s.t. its loss is n

- At round 1 the environment sets $y_1(A) = 1 - \hat{y}_1$
- At round t the prediction of the algorithm is $\hat{y}_t$ depending on $y_1(A), \dots, y_{t-1}(A)$
- At round t the environment sets $y_t = 1 - \hat{y}_t$

On the same sequence $y_1(A), \dots, y_n(A)$ at least one of the expert provides a cumulative loss smaller than $\frac{n}{2}$

Deterministic Algorithm Regret

Theorem

Any deterministic algorithm A algorithm applied to the discrete prediction problem has a worst-case regret:

$$R_n(A) \geq \frac{n}{2}$$

Solution: resort to randomization

Basic idea: we use the EWA forecaster and the weights as a probability distribution

We call this Randomized EWA forecaster (REWA)

Equivalent Problem

Let us define the following continuous prediction problem:

- the outcome space $Y' = Y \times D'^N$
- the decision space $D' = \{p \in [0,1]^N : \sum p_i = 1\}$
- the experts $f'_{i,t} = (0, \dots, 0, 1, 0, \dots, 0)$ in the i-th position of dimension D
- the performance function $l'\big(p, (y, f_1, \dots, f_N)\big) = \sum_{i=1}^{N} p_i \; l(f_i, y)$ is now convex

At each round we predict with I_t drawn from the distribution $\hat{p}_1, \dots, \hat{p}_N$ and we have:

$$E[l(I_t, y_t)] = \sum_{i=1}^{n} \hat{p}_i l(f_i, y_t) = l'\big(\hat{p}, (y, f_1, \dots, f_N)\big)$$

in expectation we have the same loss of the original EWA

REWA Regret

Theorem

The EWA algorithm applied to a continuous prediction problem with N experts and with parameter η has regret:

$$R_n(EWA) \leq \frac{logN}{\eta} + \frac{\eta n}{8}$$

Theorem

The REWA algorithm applied to a discrete prediction problem with N experts and with parameter η has regret:

$$E[R_n(REWA)] \leq \frac{logN}{\eta} + \frac{\eta n}{8}$$

High Probability Analysis

This means that on some specific runs the REWA is performing arbitrarily bad

Theorem (Hoeffding-Azuma Bound)

Given a set of n random variables $X_1, \dots, X_n$ defined over the support [0,1] the following holds:

$$P\left(\sum_{t=1}^{n} X_t - \sum_{t=1}^{n} E[X_t] > \epsilon\right) \leq e^{-\frac{2\epsilon^2}{n}}$$

Applying it to $\mathrm{X_t} = l(I_t, y_t)$, with $\delta = e^{-\frac{2\epsilon^2}{n}}$, we have:

$$P\left(\sum_{t=1}^{n} l(I_t, y_t) - \sum_{t=1}^{n} E[l(I_t, y_t)] > \sqrt{\frac{nlog1/\delta}{2}}\right) \leq \delta$$

High Probability Bound

Merging the regret bound in expectation and the high probability bound we have:

Theorem

The REWA algorithm applied to a discrete prediction problem with N experts and with parameter $\eta = \sqrt{\frac{8 log N}{n}}$ satisfies:

$$R_n(REWA) \leq \sqrt{\frac{n log N}{2}} + \sqrt{\frac{n log 1/\delta}{2}}$$

with probability at least $1 - \delta$

FL Revisited

Are we still able to use FL for this problem? No, we would have the same problems since it is a deterministic algorithm

Follow The Perturbed Leader (FPL)

at each round t

collect the experts' predictions $f_{i,t}$

predict $\hat{y}_t = f_{E,t}$ where $E = \arg\min_{i\in\{1,\dots,N\}} \left(\sum_{s=1}^{t-1} l(f_{i,s}, y_s) + Z_{i,t}\right)$

observe y_t and suffer loss $l(\hat{y}_t, y_t)$

Where $Z_{i,t}$ are realizations of i.i.d. random variables

FPL Regret Bound

Theorem

The FPL algorithm applied to a discrete prediction problem with N experts and with parameter satisfies:

$$R_n(FPL) \leq 2\sqrt{nN} + \sqrt{\frac{nlog1/\delta}{2}}$$

With probability at least $1 \ - \delta$

Worse result in terms of number of number of experts N than REWA

We can restore the order of $logN$ choosing carefully the random variables $Z_{i,t}$ using a two-sided exponential distribution

$p(z) = \left(\frac{\eta}{2}\right)^N e^{-\eta|z|}$ (with $\eta > 0$)

Expert Learning

Infinite Number of Experts

Portfolio Optimization

Sequential investment problem:

- Given budget W_0, invest it over a set of N different stocks
- At each round, I am required to choose a distribution $\widehat{\boldsymbol{y}_t} = (\hat{y}_{1,t}, \dots, \hat{y}_{N,t})$ of the budget over the available stocks
- The environment chooses a vector $\boldsymbol{y}_t = (y_{1,t}, \dots, y_{N,t})$ telling us the stock prices
- At the end of n investment steps we have a total wealth of:

$$W_n = \sum_{i=1}^{N} \widehat{y}_i\, W_{n-1} y_i = W_0 \prod_{t=1}^{n} \widehat{\boldsymbol{y}_t}^\top \boldsymbol{y}_t$$

Regret for Portfolio Optimization

Maximize W_n is equivalent to minimize $-\log W_n$, now interpreted as loss:

$$-\log W_n = -logW_0 + \sum_{t=1}^{n} -\log(\widehat{\boldsymbol{y}_t}^{\top}\boldsymbol{y}_t)$$

i.e., we are using a loss equal to $l(\widehat{\boldsymbol{y}_t}, \boldsymbol{y}_t) = -\log(\widehat{\boldsymbol{y}_t}^{\top}\boldsymbol{y}_t)$

The regret against the best constant expert becomes:

Definition: Regret

$$R_n(A) \coloneqq \sum_{t=1}^{n} -\log(\widehat{\boldsymbol{y}_t}^{\top}\boldsymbol{y}_t) \; - \; \min_{\boldsymbol{y}^* \in \Delta^N} \sum_{t=1}^{n} \log({\boldsymbol{y}^*}^{\top}\boldsymbol{y}_t)$$

where $\boldsymbol{y}^*$ is the best constantly rebalanced portfolio

Portfolio Optimization Problem

- the outcome space Y is discrete ($|Y| > 2$)
- the decision space $D = \Delta^N$
- the performance function $l(\widehat{\boldsymbol{y}_t}, \boldsymbol{y}_t) = -\log(\widehat{\boldsymbol{y}_t}^\top \boldsymbol{y}_t)$

Can we use the EWA forecaster? We need to define its continuous version

Define $w_t(\boldsymbol{a}) = \exp(-\eta \sum_{s=1}^{t-1} -\log(\boldsymbol{a}^\top \boldsymbol{y}_t))$ and $z_t = \int_{\boldsymbol{a} \in \Delta^N} w_t(\boldsymbol{a})\, d\boldsymbol{a}$

at each round t

predict $\hat{y}_t = \int_{\boldsymbol{a} \in \Delta^N} \frac{w_t(\boldsymbol{a})}{z_t}\, \boldsymbol{a}\, d\boldsymbol{a}$

observe y_t and suffer loss $-\log(\widehat{\boldsymbol{y}_t}^\top \boldsymbol{y}_t)$

update weights $w_t(\boldsymbol{a})$ and W_t

Regret of the CEWA

Theorem

The CEWA algorithm applied to an online portfolio optimization problem with N stocks and with parameter $\eta = 2\sqrt{\frac{2N\log n}{n}}$ has regret:

$$R_n(CEWA) \leq 1 + \sqrt{\frac{Nn\log n}{2}}$$

Sublinear but not satisfactory

What if we use a strategy which tries to follow the leader i.e. the combination of stocks providing, so far, the best wealth

Universal Portfolio

Weighted average approach

Define the wealth of a constant strategy $W_n(\boldsymbol{a}) = W_0 \ \prod_{t=1}^{n-1} \boldsymbol{a}^\top \boldsymbol{y_t}$

at each round t

predict $\hat{y}_t = \dfrac{\int_{\boldsymbol{a} \in \Delta^N} \boldsymbol{a} \, W_t(\boldsymbol{a}) \, d\boldsymbol{a}}{\int_{\boldsymbol{a} \in \Delta^N} W_t(\boldsymbol{a}) \, d\boldsymbol{a}}$

observe y_t and suffer loss $-\log(\widehat{\boldsymbol{y_t}}^\top \boldsymbol{y_t})$

Theorem

The UP algorithm applied to an online portfolio optimization problem with N stocks has regret:

$$R_n(UP) \leq (N-1) \log(n+1)$$

From Model to Real-World

Even if this model is quite general it does not take into account some of the peculiar aspects of reality:

- When exchanging stocks also implies to have transaction costs, therefore one might need to limit the variation of the strategy over time
  - Vittori et al., Dealing with Transaction Costs in Portfolio Optimization: Online Gradient Descent with Momentum, ICAIF, 2020
  - Das et al., Online Lazy Updates for Portfolio Selection with Transaction Costs, AAAI, 2013
- Using online learning might provide low performance at the beginning of the learning period, therefore its application might scare the investors
  - Bernasconi et al., Conservative Online Convex Optimization, ECML, 2020

Lecture Recap

We have guarantees on the loss we incur in a set of different problems when using specific algorithm:

- Prediction problem with a finite number of experts
- Classification problem over finite number of classes
- Prediction problem with an infinite number of experts

Depending on the loss function we suffer, we might want to use different algorithms

On some specific classes of problems we are sure no other algorithm performs as good as the algorithms described here

Bibliography

Cesa-Bianchi, Nicolo, and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

Shalev-Shwartz, Shai. "Online learning and online convex optimization." *Foundations and Trends® in Machine Learning* 4.2 (2012): 107-194.

Next Lecture

- Switch to settings in which the feedback is only limited
- Analysis of the cases in which the environment is stochastic but we want to operate in an online manner