POLITECNICO DI MILANO
Mathematical Eng Master Degree
Dipartimento di Mate

Il Grinch Aveva Ragione

Artificial Intelligence and Robotic Laboratory
of Politecnico di Milano

Supervisor: Prof. Marcello Restelli

Co-supervisors: Francesco Trovò Ph.D., Edoardo Vittori,

Master thesis of:
Martino Bernasconi de Luca, π

Academic Year 2018-2019

Contents

List of Figures

Chapter 1

Introduction

Classical investment techniques for the portfolio management problem derive from the knowledge of the statistical distribution of the assets return. Then, once the statistical model has been chosen, the problem get solved by optimizing the expected value of the utility of some random variable (usually accounting for the trade-off between risk and return), that describes the value of the portfolio in some fixed time in the future. This line of thinking has been proposed and sustained by Markovitz, Samuelson, Fama ecc... , and it is now called Modern Portfolio Theory (MPT).

This approach is known to be very susceptible to the errors in the modelling of the random variable that model the asset return. In fact, it is known that the markets have a non stationary behaviour, which means that every statistical assumption is ephemeral and unreliable. and they are usually referred to to backward looking, i.e. that they optimize

A different approach has been originated from the fields of information theory at the Bell Labs in the 1950, from the works of Shannon, Kelly and Cover. This methods were first included in the classical portfolio theory framework, under the name of Capital Growth Theory (CGT) [Hakansson et al., 1995] and then got included in the machine learning literature under the framework of Online Game Playing. Only recently this field has been taken into the Online Optimization This formulation has very interesting properties such as stability in a game theory fashion (equilibrium) and robustness versus adversarial manipulation.

One of the strongest points in favor of this techniques are the strong theoretical guarantees that algorithms developed under this framework can give. This guarantees come from the game theory concept of Regret, which is a form of dissatisfaction originated from having taken an action, instead of another action.

Principal in this thesis will be the extension of the modelling of the financial applications of this methodologies to the presence of transaction costs and to provide strong theoretical assurance even in the presence of transaction costs. In fact in many financial situations, transaction costs are not modelled and

3

Chapter 2

Online Learning

Online Learning is a theoretical framework to formalize a sequential decision problem in which an agent has to take consecutive actions in an environment. Every time the agent takes an action, the environment returns a loss signal (or reward depending on the convention on the sign). This framework is similar to other sequential decision problems such as Reinforcement Learning [Sutton et al.,], with the exception that the loss function is decided by an adversary which has complete knowledge of your strategy in advance, rather that be described by a stochastic probability kernel. The purpose of this section is to present the general framework of Online Game Playing and to introduce the notation necessary for the development of the theory. We will define formally the framework of Online Learning with Expert Advice, which is one the most studied framework of Online Learning, due to its ability to include many other frameworks, such as Multi Armed Banditor Online Convex Optimization. Then we will present the concept of *regret* and present the relationship of Online Learning to classical repeated games, a classical framework coming from the field of Game Theory. Modern finance has more and more the need for a Game Theoretic approach, this is evident when one looks at the field of on venue market making, that can be modeled naturally as a repeated game, or in merger and acquisition that can be modeled as a normal form game. Finally we will introduce Online Convex Optimization as a special case of Online Learning with expert advice and its interesting relationship to theoretical statistical learning. The choice of this path, from Online Learning to Online Convex Optimization, has been done to show how general and powerful Online Learning is in its simplicity, and why Online Convex Optimization is the most suitable framework to present our contribution to Online Portfolio Selection, that will be presented in later chapters.

Francesco dice che è sbagliato.

define

maybe add DAgger

manca coeasion OL->finance

add ref e aggiustare

ref chapter ...

In fact, even if we will focus on the portfolio problem, the apparently simple formulation of this framework is capable to encompass many other applications and problems, such as network routing [Belmega et al., 2018] and dark pool order allocation [Agarwal et al., 2010]. A thorough dissertation of the techniques that have been developed in the field of Online Learning can be found in [Cesa-Bianchi and Lugosi, 2006].

2.1 Online Learning

Definition 2.1.1. *(Online Game Playing). Let $\mathcal{Y}$ be the outcome space, $\mathcal{D}$ the prediction space and $f : \mathcal{D} \times \mathcal{Y} \to \mathbb{R}$ is a loss function, an Online Game is the following sequential game played by the forecaster $\mathcal{A}$ and the environment:*

For each round $t \in 1, 2, \ldots$

1. *The learner $\mathcal{A}$ chooses an element of the decision space $x_t \in \mathcal{D}$.*
2. *The environment chooses the element $y_t \in \mathcal{Y}$, and subsequently determines the loss function $f(\cdot, y_t)$.*
3. *The agent $\mathcal{A}$ incurs in a loss $f(x_t, y_t)$.*
4. *The agent updates its cumulative losses $L_t = L_{t-1} + f(x_t, y_t)$ with $L_0 = 0$*

In Online Learning an agent $\mathcal{A}$ has to guess the outcome y_t based on a the past outcomes $y_1, y_2, \ldots, y_{t-1}$ of some events that are in the outcome space $\mathcal{Y}$, at each time step she will play (sometimes we will also say *predict*) x_t, that is an element of the prediction space $\mathcal{D}$, and the environment will choose a loss function $f(\cdot, y_t)$ by determining the outcome y_t. Sometimes it is not important to know the exact outcome of the round and so we can identify the function $f(x, y_t)$ with $f_t(x)$. The agent $\mathcal{A}$ is essentially the identification of the functions that maps the history of past outcomes to the new prediction:

$$\mathcal{A} \equiv \{h_{t-1} := (y_1, \ldots, y_{t-1}) \longmapsto x_t\}_{t \geq 1}$$

The simplest case is for $\mathcal{Y} = \mathcal{D}$ and both of finite cardinality, meaning that there are only a finite number of actions that the agent $\mathcal{A}$ can choose from. We will sometimes refer to the environment defined in Section 2.1.1 as "adversarial", since no stochastic characterization is given to the outcome sequence y_t and the analysis of the regret is done assuming a worst case

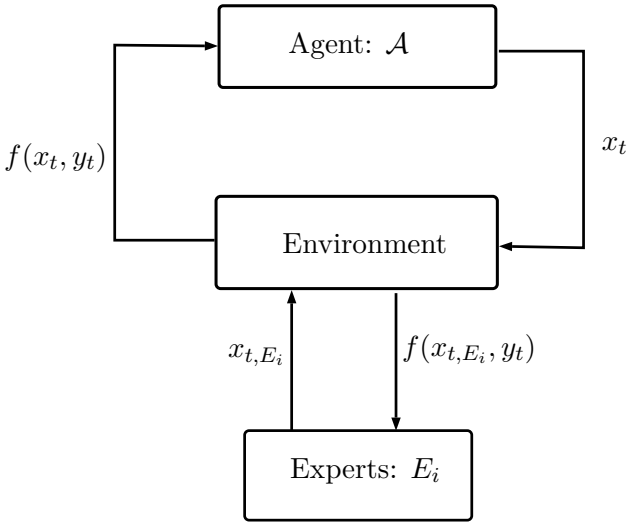

Figure 2.1: Online Learning with Expert Advice as Multi Agent-Environment interaction.

???

scenario. Since the adversary knows the prediction x_t, before deciding the outcome y_t, absolute minimization of the loss is an hopeless task and so we have to set an easier task. We will also present the counterexample to why the absolute minimization of the loss is an hopeless task, and present the adapt minimal framework to successful Online Learning in Adversarial Environment.

2.1.1 Regret and Experts

We stated that the objective of absolute loss minimization is hopeless in an adversarial framework, as the adversary can always choose the outcome y_t that maximizes the loss $f(x, y_t)$ regardless of the decision $x \in \mathcal{D}$ taken by the learner. We shall present a simple counterexample in this setting.

Take $\mathcal{D}$ as a space of binary outcomes, *i.e.* $|\mathcal{D}| = 2$ absolute loss as $f(x, y) = |x - y|$. Since the adversary plays after the learner $\mathcal{A}$, it can make the loss of the learner $L_T = T$ by outputting the bit non predicted by the learner. Notice that no assumption has been made on the strategy followed by the learner $\mathcal{A}$. From this example it is clear that the learner has to set a less ambitious goal.

We do so by extending the theoretical formulation in Section 2.1 by including a set $\mathcal{E}$ of other players, this setting is called "prediction with expert advice". At each time step of the prediction game, each expert $e \in \mathcal{E}$, predicts an element $x_{t,e} \in \mathcal{D}$, and incurs in a loss $f(x_{t,e}, y_t)$, just as the agent

Spiegare figure: Non l'ho già fatto?

$\mathcal{A}$, creating a general multi-agent interaction as in Figure 2.1. Now the goal that the learner sets itself to solve is to obtain small losses with respect to the best expert in the class $\mathcal{E}$. This concept is captured by the definition of regret. Formally, we define the regret $R_{T,e}$ for the agent $\mathcal{A}$ with respect to expert $e \in \mathcal{E}$ (assumed finite for the moment) as follows:

$$R_{T,e} = L_T - L_{T,e} \tag{2.1}$$

The regret observed by the agent $\mathcal{A}$ with respect to the entire class of experts $\mathcal{E}$ is defined as:

$$R_T = \max_{e \in \mathcal{E}} R_{T,e} = L_T - \min_{e \in \mathcal{E}} L_{T,e}. \tag{2.2}$$

The task the agent $\mathcal{A}$ is set to solve is to find a sequence x_t function of the information obtained up to the time t in order to obtain small regret y_T with respect to any sequence $y_1, y_2, \ldots$ chose by the environment.

In particular we aim to achieve sub-linear regret $y_T = o(T)$, meaning that the per-round regret y_T/T will asymptotically vanish:

$$y_T = o(T) \implies \lim_{T \to \infty} \frac{y_T}{T} = 0, \tag{2.3}$$

where $o(T)$ is the space of sub-linear affine functions. A strategy $\mathcal{A}$ that attains sub-linear regret is called *Hannan-Consistent* [Hannan, 1957].

The regret is a measure of the distance between our online performance and the best offline (in retrospect) performance among the expert class $\mathcal{E}$, this is also called *external regret* since it is compared to the external set of experts $\mathcal{E}$. A surprising fact is even that such algorithms do even exist. Indeed a first result is that in general there are no Hannan Consistent strategies, and just introducing the concept of regret is not enough for successful Online Learning:

A first simple counterexample can be found in [Cover, 1966]. If the decision space $\mathcal{D}$ is finite then there exists a sequence of loss function such that $R_T = \Omega(T)$. Again take $\mathcal{D}$ as a space of binary outcomes, absolute loss as $f(x, y) = |x - y|$, and the class of experts is composed by two experts, one predicting always 0 and the other always 1. Taking T odd, we have that the loss of the best expert is $L_{T,e} < \frac{T}{2}$, and we have already shown that the adversary can make the loss of the learner $L_T = T$. It is now evident that the regret is $R_T > T - \frac{T}{2}$, which do not allow $R_T/T \to 0$. This argument is easily extended in the case of any finite decision space $\mathcal{D}$.

In order for the learner to achieve sub-linear regret is to randomize its predictions, the learner, at each turn t, holds a probability distribution on

the decision space and plays x_t according to this distribution. Clearly the adversary has knowledge of the probability distribution of the learner $\mathcal{A}$, but has no knowledge of the random seed used by the agent $\mathcal{A}$, *i.e.* does not know the actual decision taken according to the distribution held by the agent. If the original decision space was $\mathcal{D}$ with $|\mathcal{D}| = N$ after the randomization of the decision, we effectively transformed the decision space $\mathcal{D}$ into the $\Delta_{N-1} \in \mathbb{R}^N$ probability simplex. By doing so we are formally extending the game into its mixed extension, as will be discussed further in Section 2.2. It can be viewed also as a *covexification* of the domain, pointing to the undeniably necessity of convex geometry in this context, that will be discussed in 2.3. Therefore, from now on the domain $\mathcal{D}$ will be convex, either by the problem specification or by randomized convexification if the problem has discrete decision space.

2.1.2 Existence of No-Regret Strategies

In this section we will show the existence of Hannan-consistent strategies in the case of finite experts and provide a general form to generate sub-linear regret strategies. The general idea with a finite class of experts is given by the Weighted Average Forecaster. A natural class of algorithm to explore is the predicting as the weighted average prediction of the experts predictions, weighted on the accrued regret suffered by the agent $\mathcal{A}$:

Definition 2.1.2. *(Weighted Average Forecaster). For a finite class of experts $\mathcal{E}$ the weighted average prediction is defined as*

$$x_t = \frac{\sum_{i=1}^{N} w_{t-1,i} x_{t,i}}{\sum_{i=1}^{N} w_{t-1,i}}, \tag{2.4}$$

where $w_{t-1,i} > 0$ and $x_{t,i}$ is the prediction of expert $E_i \in \mathcal{E}$ at round t.

Since $\mathcal{D}$ is convex we have that $x_t \in \mathcal{D}$. Then it is natural to assume that the weights are a function of the cumulated regret suffered by the agent with respect to the experts, and also that the change in weight is proportional to the change in a potential function: We can generalize the simple weighted average prediction (2.1.2) in the following general form, introduced in [Cesa-Bianchi and Lugosi, 2003]:

$$x_t = \frac{\sum_{i=1}^{N} \partial_i \Phi(\mathbf{R}_{t-1}) x_{t,i}}{\sum_{i=1}^{N} \partial_i \Phi(\mathbf{R}_{t-1})}, \tag{2.5}$$

where $\Phi(\mathbf{u}) = \varphi\left(\sum_{i=1}^{N} \phi(u_i)\right)$ is a function $\Phi : \mathbb{R}^N \to \mathbb{R}^+$ defined through two increasing functions $\phi, \varphi : \mathbb{R} \to \mathbb{R}^+$, $\varphi, \phi \in \mathcal{C}^2(\mathbb{R})$ and concave and convex, respectively and $\mathbf{R}_T = (R_{T,1}, \ldots, R_{T,N})$. By specializing the two functions φ, ϕ we can derive most of the algorithm for dealing with prediction under expert advice. The reasons behind the general form of Equation (2.5) and an extended discussion can be found in [Hart and Mas-Colell, 2001], [Cesa-Bianchi and Lugosi, 20 and [Blackwell et al., 1956], but the general idea is that the form of Equation (2.5) has the following property:

Theorem 2.1.1. *If x_t is given by Equation (2.5) and the loss $f(\cdot, y)$ is convex in the first argument then the instantaneous weighted regret satisfies:*

$$\sup_{y_t \in \mathcal{Y}} \sum_{i=1}^{N} [f(x_t, y_t) - f(x_{t,i}, y_t)] \partial_i \Phi(\mathbf{y}_{t-1}) \leq 0$$

.

Proof. By convexity of $f(\cdot, y_t)$ we have that

$$f(x_t, y_t) \leq \frac{\sum_{i=1}^{N} \partial_i \Phi(\mathbf{R}_{t-1}) f(x_{t,i}, R_t)}{\sum_{i=1}^{N} \partial_i \Phi(\mathbf{R}_{t-1})}, \forall y_t \in \mathcal{Y} \tag{2.6}$$

And since $\Phi(\mathbf{x}) = \varphi\left(\sum_{i=1}^{N} \phi(x_i)\right)$ we have that

$$\partial_i \Phi(\mathbf{x}) = \varphi'\left(\sum_{i=1}^{N} \phi(x_i)\right) \phi'(x_i) \geq 0$$

Hence we can rearrange the terms in Equation (2.6) to obtain the statement. □

Note that fixing the structure for the weights as in Equation (2.5) we have that $w_{t,i} \propto \phi'(R_{t,i})$ that is an increasing function in $R_{t,i}$ (since ϕ is convex and increasing) that essentially states that we are increasing the probability of playing actions on which we saw high regret $R_{t,i}$.

check

Definition 2.1.3. *The exponentially weighted algorithm is* (2.5) *with* $\varphi(x) = \frac{1}{\eta} ln(x)$ *and* $\phi(x) = e^{\eta x}$ *giving weights of the form* $w_{t-1,i} = e^{\eta y_{t-1,i}} / \sum_{j=1}^{N} e^{\eta y_{t-1,j}}$

e allora show

It can be shown that the algorithm defined by the update rule in Equation (2.1.3), and for a convex loss function $f(\cdot, y_y)$, gives the following guarantee on the regret:

$$R_T \leq \frac{log(N)}{\eta} + \frac{T\eta}{8} \tag{2.7}$$

,

By choosing $\eta = O\left(\sqrt{\frac{1}{T}}\right)$ we obtain a sub-linear regret $R_T = \mathcal{O}(\sqrt{T})$.

2.1.3 Experts

The theoretical framework described in Section 2.1 is very general and most suited for a game theory analysis of the problem. This help us describe many other frameworks, such as Online Optimization, or Multi Armed Bandit (MAB) as embedded into a Game Playing framework with expert advice. It can then be specialized by fixing many elements of the definition, in order to be applied to the specific problem we are willing to solve. For instance, the class of experts $\mathcal{E}$ is most of the time completely fictitious, meaning that the experts are not real players of the game, but most of the time they are *simulable* meaning that the agent $\mathcal{A}$ is able to compute $x_{t,e}$ for each expert $e \in \mathcal{E}$ and most of the times the class of expert is very limited in its actions, *e.g.* $\mathcal{E}$ is the class of experts for which $x_{t,e}$ is constant in t. In this case, which is the most studied class of experts, we are basically just comparing our learner $\mathcal{A}$ to the best fixed action x^* in hindsight. This is a clairvoyant strategy that attains the minimum cumulative loss over the entire length of the game T.

2.1.4 Uncountable Experts

In the case of uncountable experts the Exponentially Averaged Prediction cannot be applied directly, but can be extended to a continuous mixture of experts predictions. More specifically we need the case of the class $\mathcal{E}$ being generated by a convex hull of a finite number of a base class of experts, $\mathcal{E}_N$. With continuous class of experts $\mathcal{E}$ defined in this way, the regret definition becomes:

$$R_T = \sup_{q \in \Delta_{N-1}} R_{T,q} := L_T - \inf_{q \in \Delta_{N-1}} L_{T,q}, \tag{2.8}$$

where $\Delta_{N-1} \subset \mathbb{R}^N$ is the N-simplex, and

$$L_{T,q} = \sum_{t=1}^{T} f(\langle q, x_{t,e} \rangle, y_t),$$

where $x_{t,e} = (x_{t,1}, \ldots, x_{t,N}) \in \mathbb{R}^N$ is the vector of expert predictions at time t.

2.1.5 Exp-Concave loss functions

It will be important for the study of Portfolio Optimization the exp-concave class of loss functions. $f(\cdot, y)$ is ν-exp concave if $e^{-\nu f(\cdot,y)}$ is concave.

Theorem 2.1.2. *The Exponentially Weighted Average forecaster, for ν-exp concave loss functions and for $\eta = \nu$ has the following property:*

$$\Phi(\mathbf{R}_T) \leq \Phi(\mathbf{R}_0)$$

where $\Phi(x) = \varphi\left(\sum_{i=1}^{N} \phi(x_i)\right)$ *is chosen as* $\varphi(x) = \frac{1}{\nu} log(x)$ *and* $\phi(x) = e^{\nu x}$

Proof. The weights are given by $w_{t-1,i} = e^{\nu y_{t-1,i}} / \sum_{j=1}^{N} e^{\nu y_{t-1,j}}$. By exp-concavity we have that

$$e^{-\nu f(x_t,y_t)} = exp\left\{-\nu f\left(\frac{\sum_{i=1}^{N} w_{t-1,i} x_{t,i}}{\sum_{i=1}^{N} w_{t-1,i}}, y_t\right)\right\} \geq \frac{\sum_{i=1}^{N} w_{t-1,i} e^{-\nu f(x_{t,i},y_t)}}{\sum_{i=1}^{N} w_{t-1,i}} \tag{2.9}$$

this can be rewritten as

$$\sum_{i=1}^{N} e^{\nu y_{t-1,i}} e^{\nu[f(x_t,y_t)-f(x_{t,i},y_t)]} \leq \sum_{i=1}^{N} e^{\nu y_{t-1,i}} \tag{2.10}$$

Applying $\varphi(x) = \frac{1}{\nu} log(x)$ to both sides of equation (2.10) we obtain that

$$\Phi(\mathbf{R}_t) \leq \Phi(\mathbf{R}_{t-1})$$

that prove the thesis. □

The case of exp concave functions is very special, since we can obtain Theorem 2.1.2 that can be used to prove regret bounds very easily:

$$R_T \leq \frac{1}{\eta} \log\left(\sum_{i=1}^{N} e^{\nu R_T,j}\right) = \Phi(\mathbf{R}_T) \leq \Phi(\mathbf{R}_0) = \frac{logN}{\eta} \tag{2.11}$$

The case of exp-concave losses is also useful for the case of uncountable experts sketched in Section 2.1.4. This formulation will be of central importance for the portfolio optimization problem.

It is natural to extend the Exponential Weighted Majority algorithm described by equation (2.1.2) into its continuous case by:

$$x_t = \frac{\int_{\Delta_{N-1}} w_{q,t-1}\langle q, x_{t,e}\rangle dq}{\int_{\Delta_{N-1}} w_{q,t-1} dq} \tag{2.12}$$

Theorem 2.1.3. *(Mixture forecaster for exp-concave losses). Choosing* $w_{q,t-1} = exp\left\{-\eta \sum_{s=1}^{t-1} f(\langle q, x_{t,e}\rangle, y_s)\right\}$ *in Equation* (2.12)*, for a bounded* ν*-exp concave loss function* $f(\cdot, y)$*, we obtain*

$$R_T \leq N\nu\left(log\left(\frac{\nu T}{N}\right) + 1\right)$$

Even in the case of uncountable many experts, exp-concavity of the loss function gives a better convergence rate of $\mathcal{O}(\log T)$then the exponentially weighted algorithm in Equation (2.7), which is $\mathcal{O}(\sqrt{T})$.

2.2 Regret Minimization in Games

In this section we explore the connection of the framework of Section 2.1 into a more classical repeated game framework. In the previous Section we looked at the adversary as a black box, without any specific model in mind. The reason of this chapter is to clarify its role as a player in the game and to show the game theoretical properties of Hannan-consistent agents. Since in Online Learning the convention is to speak about losses, we shall speak about losses (minimization) also in the classical definitions of game theory instead of payoffs (maximization).

Forse da togliere tutto questo sub-chpt

Definition 2.2.1. *(Strategic Form* K*-Player Game). A Strategic form* K*-player game is t a tuple* $\langle \mathcal{K}, \{X_i\}_{i\in\mathcal{K}}, \{l_i\}_{i\in\mathcal{K}}\rangle$ *where*

1. $\mathcal{K} = \{1, \dots, K\}$ *is the finite set of players*
2. X_i *is the set of actions available to player* $i \in \mathcal{K}$
3. $l_i : \bigotimes_{k=1}^{K} X_i \to \mathbb{R}$ *is the loss observed by player* $i \in \mathcal{K}$

The game is called finite if $|X_i| < +\infty$ *for all* $i \in \mathcal{K}$.

2.2.1 Mixed extension

As in Section 2.1 we saw that it is impossible to obtains sub-linear regret in adversarial environment with finite decision space $\mathcal{D}$. A first step to solve this has been the *randomized convexification*, where finite action spaces are extended into convex sets, given by their probability simplex, also losses are to be interpreted as expected losses when the mixed extension is applied to the formal game. More formally:

Definition 2.2.2. *(Mixed-extension for finite games). A finite game* $\langle \mathcal{K}, \{X_i\}_{i\in\mathcal{K}}, \{l_i\}_{i\in\mathcal{K}}\rangle$ *can be extended into the game* $\langle \mathcal{K}, \{\tilde{X}_i\}_{i\in\mathcal{K}}, \{\tilde{l}_i\}_{i\in\mathcal{K}}\rangle$

1. $\tilde{X}_i = \Delta_{|X_i|-1} \subset \mathbb{R}^{|X_i|}$ *for all* $i \in \mathcal{K}$
2. $\tilde{l} : \bigotimes \tilde{X}_i \to \mathbb{R}$ *is defined as*

$$\tilde{l}(x_1, \dots, x_K) = \sum_{i_1=1}^{N} \cdots \sum_{i_K=1}^{N} p_{i_1} \dots p_{i_K} l(i_1, \dots, i_K)$$

Due to the impossibility result of Cover [Cover, 1966], we have to work with the mixed extension formulation of the game. So from now on we take this step implicitly. The taxonomy of game definition is quite extended and complex, we will focus on non-cooperative games since they are closely related to the setting tacked in the Online Learing field. More specifically, we will need the model for *Zero Sum Game*.

cita qualcuno

Definition 2.2.3. *(2-Player Zero-Sum Game). A Zero Sum game is a tuple* $\langle \{X_1, X_2\}, l : X_1 \times X_2 \to \mathbb{R}\rangle$. *As in Definition 2.2.1* X_1, X_2 *are the action spaces for Player* 1 *(row player) and Player 2 (columns player) respectively and* $l(x_1, x_2)$ *for* $x_i, x_2 \in X_1 \times X_2$*, represents the losses for Player* 1 *and profits for player* 2*.*

If this game is played for T turns, we can call it a repeated game, and the losses for each player will be $L_1^{(T)} = \sum_{t=1}^{T} l_i\left(x_i^{(t)}, x_2^{(t)}\right)$ and $L_2^{(T)} = -L_1^{(T)}$.

2.2.2 MinMax Consistency

The question of what guarantees does Hannan-consistent strategies bring to the game theoretical formulation of the problem, and why Online Learning is sometimes called *Learning in Games*. For such games we can define a

values for the game as:

$$V_1 = \inf_{x_1 \in X_1} \sup_{x_2 \in X_2} l(x_1, x_2) \tag{2.13}$$

$$V_2 = \sup_{x_2 \in X_2} \inf_{x_1 \in X_1} l(x_1, x_2) \tag{2.14}$$

These is the value that the players can guarantees themselves, meaning that no matter the strategy of the columns player, the row player could guarantee himself a loss of at maximum V_1, the converse holds for the row player. It can be interpreted as the minimum loss (best payoff) that player could achieve if we know that the other player would play adversarially. It is clear that $V_2 \leq V_1$. In the case the zero sum-game is a mixed extension of a finite game, then the Von Neumann theorem states that $V_1 = V_2$.

Now we will embed the framework of Online Game Playing of Section 2.1 in a two player zero sum game. Online Learning is a special form of Zero Sum Game (possibly considering its mixed extension described in Definition 2.2.1) where $X_1 \equiv \mathcal{D}$ and $X_2 \equiv \mathcal{Y}$. The loss function $l : X_1 \times X_2 \to \mathbb{R}$ can be identified by the loss $f : \mathcal{D} \times \mathcal{Y} \to \mathbb{R}$ of the Online Learning Agent $\mathcal{A}$. Now we will explore interesting properties of Hannan Consistent strategies. A surprising fact is that if the row player plays accordingly to a Hannan Consistent strategy then it achieve the value of the game V_1.

Theorem 2.2.1. *Hannan Consistent agents in Online Game Playing reach asymptotically the minmax value of the one shot game.*

$$\limsup_{T \to +\infty} \frac{1}{T} \sum_{t=1}^{T} f(x_t, y_t) \leq V_1$$

Proof. Let us suppose that player 1 plays an Hannan Consisten strategy and that $y_1, y_2, \ldots \subset \mathcal{Y}$ is a generic sequence played by the columns player.

$$\limsup_{T \to +\infty} \frac{y_T}{T} \leq 0 \tag{2.15}$$

that can be translate into

$$\limsup_{T \to +\infty} \frac{1}{T} \sum_{t=1}^{T} f(x_t, y_t) \leq \limsup_{T \to +\infty} \frac{1}{T} \inf_{x \in \mathcal{D}} \sum_{t=1}^{T} f(x, y_t) \tag{2.16}$$

Lets call $\hat{y}_T$ the empirical distribution played by player 2 up to T:

$$\hat{y}_T(j) = \frac{1}{T} \sum_{t=1}^{T} y_t$$

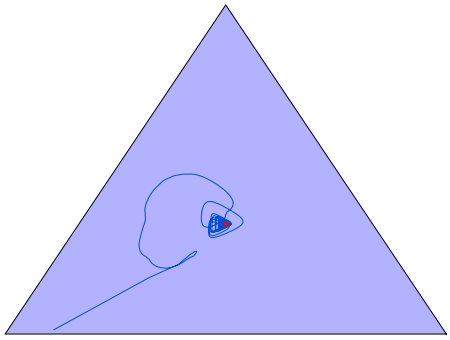

Figure 2.2: Rock Paper Scissor Dynamics Exponentially Weighted Majority

by (2.16) we just need to show $\frac{1}{T} \inf_{x \in \mathcal{D}} \sum_{t=1}^{T} f(x, y_t) \leq V$

$$\inf_{x \in \mathcal{D}} \frac{1}{T} \sum_{t=1}^{T} f(x, y_t) = \inf_{x \in \mathcal{D}} f(x, y_T) \leq \sup_{y \in \mathcal{Y}} \inf_{x \in \mathcal{D}} f(x, y) \leq V_1 \tag{2.17}$$

□

We showed that regardless of the strategy of player 2, a player playing a Hannan Consistent strategy achieves lower losses that the value of the game V_1. Clearly using a Hannan consistent strategy means that if player 2 were not adversarial, then player 1 could potentially earn a significantly higher average payoff than the value V of the game. By symmetry if both players play an Hannan Consistent strategy than they will asymptotically reach the value of the game $V = V_1 = V_2$.

2.3 Online Convex Optimization for Regret Minimization

Let's compare this framework to an apparently unrelated problem, namely optimization, this will be the most suited framework to embed the Online Portfolio Optimization Problem. In online optimization an agent $\mathcal{A}$ is set to optimize a sequence of functions $f_t(x)$ where usually $f_t : \mathcal{X} \to \mathbb{R}$ is a real valued function from the set $\mathcal{X} \subset \mathbb{R}^n$. IN Online Convex Optimization literature, some times the loss functions are identified as $f(x, y_t) \equiv f_t(x)$. The decision space $\mathcal{D}$ is assumed to be convex, as the are the functions $f_t : \mathcal{D} \to \mathbb{R}$. This framework was first devised in [Zinkevich, 2003], and has been later wildly used in the machine learning community to engineer optimization procedures [Shalev-Shwartz et al., 2012].

Convexity plays an central role in most of the analysis made in Online Learning, and Online Convex Optimization. Convexity of the domain $\mathcal{D}$ and of the loss functions, $f(\cdot, r)$ bound the problem geometry and let us derive simple and efficient learning procedures. More generally in the subsequent section we will present the general learning

2.3.1 Statistical Learning and Online Learning

Now we explore the connection between the Online Optimization framework and classical concepts of classical Statistical Learning techniques. More concretely we can prove and design a whole class of algorithm that are Agnostically PAC Learnable with Online Learning Techniques. Classical statistical learning theory deals with examples (or observations) and models of the phenomena. Then it uses the model to predict the future observations [Bousquet et al., 2003]. Quite informally one could say that we are trying to infer concept from examples. A concept is a map $\mathcal{C} : \mathcal{X} \to \mathcal{Y}$, where X is the domain space and $\mathcal{Y}$ is the set of labels for the examples. We then observe a sample from an unknown distribution $\mathcal{D}$ such that $(x, y) \sim \mathcal{D}$. What we need to achieve is to learn a mapping $y : \mathcal{X} \to \mathcal{Y}$ such that the error under the distribution $\mathcal{D}$ is small. The loss function needed to define this error is not specific to the problem and can be decided by the user, this is called generalization error and, for a loss function $l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, it is defined as:

$$e(h) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[l(h(x), y)] \tag{2.18}$$

The goal for an algorithm $\mathcal{A}$ is to produce a hypothesis h with small generalization error. It is generally difficult to generalize well and how difficult is clarified by the following theorem called the *No free lunch theorem* . There

cita

quale cit?

are many variation of this theorem, there is one formulation which states that: for any learner $\mathcal{A}$ that learns an hypothesis $h : \mathcal{X} \to \{0, 1\}$, there exists a concept $\mathcal{C}$ with generalization error 0 and a distribution $\mathcal{D}$ such that the generalization error of $\mathcal{A}$ is at least $1/2 - \epsilon$ for any $\epsilon > 0$. Hence it is impossible to learn any concept in this general sense. But we can learn concept restricting the class of concepts in a hypothesis space $\mathcal{H} : \mathcal{X} \to \mathcal{Y}$. This restriction gives raise to the concept of Probably Approximately Correct (PAC) learnability.

Definition 2.3.1. *(PAC learnable). An hypothesis class $\mathcal{H}$ is PAC learnable w.r.t. the loss l if there exists a learner $\mathcal{A}$ that given a sample S_N of examples learns an hypothesis $h \in \mathcal{H}$ s.t. for all ϵ, δ there exists $N_{\epsilon,\delta}$ such that for any distribution $\mathcal{D}$ we have a generalization error $\mathbb{P}[e(h) < \epsilon] \geq 1 - \delta$*

Usually we also require that the algorithm $\mathcal{A}$ learns the concept h in polynomial time w.r.t. the parameter of the problem.

An example of such learning problems could be the classification of spam emails. In this case $\mathcal{X}$ is the vectorial representation of the text and $\mathcal{Y} = \{0, 1\}$, indicating weather or not the email it a spam or not. If we choose as a model a linear classifier then the hypothesis space is $\mathcal{H} = \{h = \mathbb{I}[\langle x, w\rangle \geq 1/2]\}$ and the loss could be chose as $l(y_1, y_2) = |y_1 - y_2|$.

PAC learnability is intuitively requiring that the there exists an hypothesis $h \in \mathcal{H}$ with near zero generalization error, otherwise the class $\mathcal{H}$ is not PAC learnable, otherwise the class $\mathcal{H}$ is not PAC learnable. But we can weaken the concept of PAC learnability by addressing directly this issue.

Definition 2.3.2. *(PAC agnostic learnable). Given the same definitions of Definition 2.3.1, an hypothesis class $\mathcal{H}$ is PAC agnostic learnable if we have a generalization error $\mathbb{P}[e(h) < \inf_{\tilde{h} \in \mathcal{H}} e(\tilde{h}) + \epsilon] \geq 1 - \delta$*

Which hypothesis spaces $\mathcal{H}$ are PAC learnable (agnostically or not) is an open and complex issue, but the case for convex hypotheses class $\mathcal{H} \subset \mathcal{R}$ can be solved by Online Learning techniques, showing the versatility of the methods. Moreover approach to prove such theorem gives an constructive methodology to solve agnostic PAC learnable problems.

Theorem 2.3.1. *For every hypothesis class $\mathcal{H}$ and bounded loss function $l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, for which does exists a low regret algorithm $\mathcal{A}$, then the problem is angostic PAC learnable. In particular this conditions are satisfied if the hypotesis space $\mathcal{H}$ and the loss function l are convex.*

Proof. (Sketch). Initialize the learner with the hypothesis $h_0 = \mathcal{H}$. For every iteration $t \leq T$: observe a sample $(x_t, y_t) \sim \mathcal{D}$ and a loss function $l_t := l(h_t(x_t), y_t)$. Then update the hypotesis $h_{t+1} = \mathcal{A}(l_1, \ldots, l_t)$.

At $t = T$ return $\bar{h} = \frac{1}{T}\sum_{t=1}^{T} h_t \in \mathcal{H}$.

The proof then continues by defining the random variable $X_T^{(1)} = \sum_{t=1}^{T} e(h_t) - l(h_t(x_t), y_t)$ this is a martingale and $\mathbb{E}[X_T^{(1)}] = 0$. Moreover $|X_T^{(1)} - X_{T-1}^{(1)}| < K$ since the loss function f is bounded. We can normalize the losses so that $K = 1$, and then apply the Azuma martingale inequality $\mathbb{P}[X_T^{(1)} > c] \leq e^{-\frac{c^2}{2T}}$.

For an appropriate choice of c we get

$$\mathbb{P}\left[\frac{1}{T}\left[\sum_{t=1}^{T} e(h_t) - l(h_t(x_t), y_t)\right) > \sqrt{\frac{2\log(\delta/2)}{T}}\right] \leq \delta/2 \tag{2.19}$$

defining $h^* = \arg\inf_{h\in\mathcal{H}} e(h)$ and $X_T^{(2)} = \sum_{t=1}^{T} e(h^*) - l(h^*(x_t), y_t)$ we can obtain

$$\mathbb{P}\left[\frac{1}{T}\left(\sum_{t=1}^{T} e(h^*) - l(h^*(x_t), y_t)\right) < -\sqrt{\frac{2\log(\delta/2)}{T}}\right] \leq \delta/2 \tag{2.20}$$

By the definition of regret y_T we obtain

$$\frac{1}{T}\sum_{t=1}^{T} e(h_t) - e(h^*) = y_T/T + X_T^{(1)} - X_T^{(2)} \tag{2.21}$$

and from inequalities (2.19), (2.20) and from Equation (2.21) we have:

$$\mathbb{P}\left[\frac{1}{T}\sum_{t=1}^{T} e(h_t) - e(h^*) > \frac{y_T}{T} + 2\sqrt{\frac{2\log(\delta/2)}{T}}\right] \leq \delta \tag{2.22}$$

Now simply thanks to the linearity of the error operator $e : \mathcal{H} \to \mathbb{R}$ we have that

$$\mathbb{P}\left[e(\bar{h}) < e(h^*) + y_T/T + 2\sqrt{\frac{2\log(\delta/2)}{T}}\right] \leq 1 - \delta$$

and since $y_T/T \to 0$ we can find $\tilde{T}$ large enough such that the thesis is verified. □

This result has been presented since it is useful to prove the general behavior of Hannan consistent strategies in environments driven by a stationary distribution.

Chapter 3

Information, Prediction and Investing

In Chapter 2 we described at a high level the framework of Online Learning in Adversarial environment. Now we draw its connections with predictions and investments. It surly seems counter intuitive to speak about predictions in an adversarial framework, since we are used to think about predictions only of stochastic processes, but the way to think about predictions in adversarial environments is to think about probability assignment and empirical frequencies. The root of this formulation are to be traced back to the Bell Laboratories in the '50, from works of Kelly [Kelly Jr, 2011], linking sequential betting and concept from information theory [Cover and Thomas, 2012]. This connection is of primary importance to understand sequential investing as an instance of sequential decision problem. We first draw the parallelism between probability assignment over discrete events and Online Learning and then extend the discussion to sequential investments.

3.1 Probability assignment

The decision space $\mathcal{D}$ in the case of finite N possible bets is the $\Delta_{N-1} \subset \mathbb{R}^N$ probability simplex while the outcome $\mathcal{Y}$ space is the set $\{1, \ldots, N\}$, representing the winning bet at each turn. The loss function $f(x, y)$ should have these natural properties: low when x_y 1 and high when x_y 0 where x_y is the probability assigned to the outcome y. The inverse log-likelihood seems a reasonable proposal, simply because the multiplicative additive property of the logarithm but has also a deeper connection to information that we will discuss later on:

Definition 3.1.1. *(Self Information Loss). In the sequential probability*

assignment problem the loss function $f(x, y)$, $x \in \Delta_{N-1}$ *and* $y \in [1, \dots, N]$ *is defined as*

$$f(x, y) = -\log\left(x^{(y)}\right)$$

where $x^{(y)}$ *is the probability assigned to outcome* $y \in \mathcal{Y}$.

In the case of simulable experts, the prediction x_t of the agent is a function of the history of outcomes $y^{t-1} := \{y_1, y_2, \dots, y_{t-1}\}$.

The cumulative loss for the agent $\mathcal{A}$ is then given by

$$L_T = \prod_{t=1}^{T} f(x_t, y_t) \tag{3.1}$$

and can be interpreted as the log liklyhood assigned to the outcome sequence y^T since

$$L_T = \sum_{t=1}^{T} f(x_t, y_t) = -\log\left(\prod_{t=1}^{T} x_t^{(y_t)}\right) \tag{3.2}$$

where we can interpret $\prod_{t=1}^{T} x^{(y_t)}$ as the probability assigned to the entire outcome sequence y^T. This is already very similar to the compression-entropy rate one encounters in a classical lossless encoder, such as the arithmetic encoder [Langdon, 1984]. We will explore the connections to information theory later on in the chapter.

ref esatta al ch

Similarly we can define the loss for an expert $e \in \mathcal{E}$ as

$$L_{T,e} = \sum_{t=1}^{T} f(x_{t,e}, y_t) = -\log\left(\prod_{t=1}^{T} x_{t,e}^{(y_t)}\right) \tag{3.3}$$

and the regret for each expert $e \in \mathcal{E}$ is defined as

$$R_{T,e} = L_T - L_{T,e} = \log\left(\prod_{t=1}^{T} x_{t,e}^{(y_t)} / \prod_{t=1}^{T} x_t^{(y_t)}\right) \tag{3.4}$$

and the regret w.r.t. a generic class $\mathcal{E}$ of experts is defined as

$$R_T = \sup_{e \in \mathcal{E}} \log\left(\prod_{t=1}^{T} x_{t,e}^{(y_t)} / \prod_{t=1}^{T} x_t^{(y_t)}\right) \tag{3.5}$$

where the class of experts $\mathcal{E}$ can be finite or uncountable.

3.1.1 Connection to Information Theory

The link between sequential predictions and information theory has been observed in [Kelly Jr, 2011], and connects the concept of sequential betting (or predictions) and entropy.

Kelly put himself in a setting where the bettor has to predict the outcomes of binary events, given private information from an *information channel* prone to errors, the binary events pays double for a correct prediction and zero for an incorrect one. The input bits of the information channel are the correct outcomes of the binary sequential event, but they reach the end of the private channel with probability p of being correct and $q = 1 - p$ of being wrong. Clearly the optimal strategy with $p = 1$ is to bet everything on each turn reaching a final wealth of $V_T = 2^T$. In case $p < 1$ it is not clear which strategy is the best to follow, this is clearly related and still under philosophical debate as the St. Petersburg paradox [Samuelson, 1977]. Kelly propose to maximize the grow rate of the wealth., by investing a constant fraction of the current wealth. The growth rate of the wealth is defined as

$$G = \lim_{T \to +\infty} \frac{1}{T} \log_2(V_T)$$

Calling $l \in [0, 1]$ the fraction of the wealth invested in the bet we have a capital after T turns of

$$V_T = (1 + l)^W (1 - l)^{T-W}$$

and the associated growth rate is

$$G = p \log_2(1 + l) + q \log_2(1 - l)$$

which is maximized for $f = p - q$ giving $G_{\max} = 1 + p \log_2(p) + q \log_2(q)$ which is the rate of transmission for the communication channel, *i.e.* the number of bits transferred for unite of time. This is the trivial case and can be extended to arbitrary odds and number of bets.

The equivalent formulation in Online Learning can be obtained by observing that $\mathcal{D} = \Delta_0$ and that we are betting a fraction l_t on the event being 0 and a fraction $1 - l_t$ on the the outcome being 1. In that case the wealth at time T will be $V_T = V_{T-1} l_t^{\mathbb{I}_{y_T=0}} (1 - l_t)^{\mathbb{I}_{y_T=1}}$ and hence

$$log(V_T) = \sum_{t=1}^{T} \log(l_t \mathbb{I}_{y_t=0} + (1 - l_t)\mathbb{I}_{y_t=1}) \tag{3.6}$$

which is equivalent to defining the cumulative loss

$$L_T = -log(V_T) = \sum_{t=1}^{T} -\log(l_t \mathbb{I}_{y_t=0} + (1 - l_t)\mathbb{I}_{y_t=1})$$

which is equivalent to the loss defined in Equation (3.2).

By defining the growth rate at T as $G_T = \frac{1}{T}\log_2(V_T)$ we can observe that $L_T = TG_T\log(2)$ and so a learner $\mathcal{A}$ that obtains sub-linear regret $R_T/T \to 0$, where the expert class is composed of constant experts for which $l_t = const$, is equivalent to obtaining a growth rate $G_T \to G_{\max}$.

This draws the connection to information rate as defined by Shannon in terms of information bits and growth rate of a betting strategy, and the fact that an Hannan Consistent strategy is able to converge to the highest growth rate.

3.1.2 Horse Races

In this section we will see how sequential investment is equivalent to the problem of sequential betting discussed in the previous section.

In the previous chapter we saw that how to formalize sequential betting in the simple case of doubling odds and binary outcomes into the Online Learning formulation. Now we will extend the model to account variable odds and multiple bets, and how this is connected to investing.

Let us model horse races as a sequential betting system, in which we have N horses each paying a payoff of $o_{t,i}$ $\forall i \in 1, \dots, N$. The agent $\mathcal{A}$ splits its wealth into N separate betting by choosing an element of the simplex Δ_{N-1}.

The wealth of the agent $\mathcal{A}$ at time t will be the $V_t = V_{t-1}\langle \mathbf{x}_t, \mathbf{y}_t \rangle$, where $\mathbf{y}_t = o_{y_t}\mathbf{e}_{y_t} \in \mathbb{R}^N$, *i.e.* the basis vector with 1 as the $y_t \in 1, \dots, N$ component, which represents the winning horse for the turn, and o_{y_t} is the payout of the bet at time t, on the t_y horse winning. As we did in the previous section we can apply $-\log(\cdot)$ so that we can embed the problem into an Online Learning framework. By defining

$$L_T = -\log(V_T) = -\log(V_{T-1}) - \log(\langle \mathbf{x}_t, \mathbf{y}_t \rangle),$$

that implies

$$L_T = \sum_{t=1}^{T} -\log(\langle \mathbf{x}_t, \mathbf{y}_t \rangle) \tag{3.7}$$

we obtain exactly the same formulation presented at the beginning of the chapter. Moreover, we can note that the regret R_T does not depend on depend on the value of the payout o_{y_t}.

We saw in Section 2.1.4 that Theorem 2.1.3 assures that we have a sublinear regret $R_T = \mathcal{O}(\log T)$, in case that the expert class $\mathcal{E}$ is being generated by the convex hull of finite basic experts $\mathcal{E}_N$, which in this case

can be taken as the the N experts always predicting $\mathbf{x}_{t,j} = \mathbf{e}_j, \forall j \in 1, \dots, N$. The convex hull generated by $\mathcal{E}_N$ is then composed by experts predicting a constant element of the simplex $\mathbf{x}_{t,e} = \mathbf{x}_e \in \Delta_{N-1}$.

Theorem 2.1.3 is stating that we can obtain asymptotic wealth equivalent to the one obtained by the best expert in hindsight, for all sequences of outcomes.

A very similar formulation can be obtained for the case of sequential investments. In the case of horse races we have just one winner for each day, while in the case stock investing we have a different payout for each stock. In the following Section we will present how to model sequential decision problems in the Online Learning formulation.

3.2 Online Portfolio Optimization

We can think as portfolio allocation as a sequential betting problem. Let us imagine that there are no real life issues associated with trading costs and liquidity (they will be discussed in the following chapters) then the best strategy would be to invest at each round t the entire capital on one single stock, knowing that will be the best performance stock at round t. But assuming an adversarial environment we have to randomize our allocation, or equivalently distribute our wealth accordingly to the our randomization probabilities as in Equation (3.7).

agg ref

3.2.1 The Online Portfolio Optimization Model

The model consists in sequential wealth allocation in $N \in \mathbb{N}$ stocks for discrete rounds $t \in \{1, \dots, T\}$, where T is the investment horizon. The set of times is arbitrary, but in the literature is usually thought to be in days. The evolution of the stock $i \in 1, \dots, N$ prices at time t, $P_{t,i}$, define the price relatives $r_{i,t} = \frac{P_{i,t+1}}{P_{i,t}}$, and we can define the price relative vector at time t as $\mathbf{r}_t = (r_{1,t}, \dots, r_{N,t}) \in \mathbb{R}^N$.

An investor dividing at round t its wealth W_t into a fraction $\mathbf{x}_t \in \Delta_{N-1}$ for each asset, she will get a wealth $W_{t+1} = W_t \langle \mathbf{x}_t, \mathbf{r}_t \rangle$ at round $t+1$. As in Section 3.1.1 we can define the growth rate

$$G_T = \log(W_T) = \sum_{t=1}^{T} \log(\langle \mathbf{x}_t, \mathbf{r}_t \rangle)$$

As in the case of binary outcomes, *i.e.* horse races, we can define everything in term of losses and in an Online Learning framework, by defining

the loss $f(\mathbf{x}, \mathbf{y}) = -\log(\langle \mathbf{x}_t, \mathbf{y}_t \rangle)$ and a cumulative loss as

$$L_T = \sum_{t=1}^{T} -\log(\langle \mathbf{x}_t, \mathbf{y}_t \rangle)$$

Exactly as in the previous Section, the expert class is generated by the convex hull of the base class $\mathcal{E}_N$, composed by the experts always betting on the win of the same horse $i \in 1, \ldots, N$, or equivalently allocating all the portfolio on the same asset, at every turn. The convex hull of thi class is the class of experts $\mathcal{E}$ so that at every turn t, the expert is allocating all the wealth in the element $\mathbf{x} \in \Delta_{N-1}$. In the Online Portfolio literature this clss of allocations is called *Constant Rebalancing Portfolio.* As we shall see in the next Section Constant Rebalancing Portfolios are a very powerful class of strategies.

cite

Bibliography

[Agarwal et al., 2010] Agarwal, A., Bartlett, P., and Dama, M. (2010). Optimal allocation strategies for the dark pool problem. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 9–16.

[Belmega et al., 2018] Belmega, E. V., Mertikopoulos, P., Negrel, R., and Sanguinetti, L. (2018). Online convex optimization and no-regret learning: Algorithms, guarantees and applications. *arXiv preprint arXiv:1804.04529*.

[Blackwell et al., 1956] Blackwell, D. et al. (1956). An analog of the minimax theorem for vector payoffs. *Pacific Journal of Mathematics*, 6(1):1–8.

[Bousquet et al., 2003] Bousquet, O., Boucheron, S., and Lugosi, G. (2003). Introduction to statistical learning theory. In *Summer School on Machine Learning*, pages 169–207. Springer.

[Cesa-Bianchi and Lugosi, 2003] Cesa-Bianchi, N. and Lugosi, G. (2003). Potential-based algorithms in on-line prediction and game theory. *Machine Learning*, 51(3):239–261.

[Cesa-Bianchi and Lugosi, 2006] Cesa-Bianchi, N. and Lugosi, G. (2006). *Prediction, learning, and games*. Cambridge university press.

[Cover, 1966] Cover, T. M. (1966). Behavior of sequential predictors of binary sequences. Technical report, STANFORD UNIV CALIF STANFORD ELECTRONICS LABS.

[Cover and Thomas, 2012] Cover, T. M. and Thomas, J. A. (2012). *Elements of information theory*. John Wiley & Sons.

[Hakansson et al., 1995] Hakansson, N. H., Ziemba, W. T., et al. (1995). Capital growth theory. *Handbooks in operations research and management science*, 9:65–86.

[Hannan, 1957] Hannan, J. (1957). Approximation to bayes risk in repeated play. *Contributions to the Theory of Games*, 3:97–139.

[Hart and Mas-Colell, 2001] Hart, S. and Mas-Colell, A. (2001). A general class of adaptive strategies. *Journal of Economic Theory*, 98(1):26–54.

[Kelly Jr, 2011] Kelly Jr, J. L. (2011). A new interpretation of information rate. In *The Kelly Capital Growth Investment Criterion: Theory and Practice*, pages 25–34. World Scientific.

[Langdon, 1984] Langdon, G. G. (1984). An introduction to arithmetic coding. *IBM Journal of Research and Development*, 28(2):135–149.

[Samuelson, 1977] Samuelson, P. A. (1977). St. petersburg paradoxes: Defanged, dissected, and historically described. *Journal of Economic Literature*, 15(1):24–55.

[Shalev-Shwartz et al., 2012] Shalev-Shwartz, S. et al. (2012). Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194.

[Sutton et al.,] Sutton, R. S. et al. *Introduction to reinforcement learning*, volume 135.

[Zinkevich, 2003] Zinkevich, M. (2003). Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pages 928–936.