Learning, Regret Minimization and Option Pricing

Yishay Mansour[*]
School of Computer Science
Tel Aviv University
Tel Aviv, Israel

Abstract

We relate regret minimization to various online learning tasks, and most notable option pricing.

1 Introduction

1.1 Learning

Learning has many fascinating aspects. In general, learning would be associated with using past experience to improve future performance. The lack of "optimality" in our past performance can be attributed (theoretically) both to lack of information (regrading the environment we are facing) and to lack of knowledge (regarding how should the task be performed). We will focus on *lack of information*, and relate it to a general methodology that can (partially) resolve some of the issues. More specifically we will focus on cases where an online decision maker needs to select actions in the face of uncertainty, and use game theoretic models to describe the setting. In order to better highlight our motivation, we give a few typical examples.

Rock, Paper, Scissors: The goal of the player in this game, as with any other game, is to find a "winning strategy". Since this is a zero-sum game, one candidate strategy is the classical minimax strategy. In our example the max-min strategy guarantees a zero payoff with respect to any opponent strategy, however, one could hope to make gains against weaker opponents. Who is a "weak opponent" and what is a "reasonable gain" while playing against him would be part of the comparison between different learning strategies.

Weather Prediction: An entrepreneur likes to set up a new web site to predict whether it will rain tomorrow. Unfortunately, our entrepreneur have very little meteorological

knowledge and would like to base its prediction solely of other existing web sites. Her goal is to have one of the most accurate web sites regarding *rain prediction*. First, is this task feasible, and second, how should he go about realizing it.

Route Selection: Every morning a commuter has to drive from home to work, and needs to select between multiple travel routes. Each day he only observes the travel time on the route he actually drove (and has no information regarding alternative routes). His goal is to pick the best route possible. This is similar to the weather prediction setting, but here the commuter has only partial information regarding the outcome of alternate actions (routes).

In the next section we introduce regret minimization algorithms, and for all three examples regret minimization algorithms would be able to address the desired tasks and achieve (near) optimal performance.

1.2 Regret Minimization

The general setting under consideration has an online decision maker that selects at each time step an action (from a finite class of actions $\mathcal{A}$). The performance of the online decision maker is compared a class of algorithms $\mathcal{H}$, where the regret of the decision maker is the difference between the average payoff of the best algorithm in $\mathcal{H}$ (in hindsight) and its average payoff. Regret minimization algorithms guarantee that this difference vanishes with the number of time steps T, and is at most of the order $\sqrt{\log |\mathcal{H}|/T}$. (Many times, for simplicity, it is assumed that the class $\mathcal{H}$ is simply $\mathcal{A}$, namely each algorithm $h_a \in \mathcal{H}$ selects the action a in every time step.)

Regret minimization algorithms have very interesting additional guarantees. In a zero-sum game, such algorithms are guaranteed to approach or exceed the minimax value of the game (depending on the opponent). Many of the regret minimization algorithms guarantee that dominated actions would be played only rarely. For a more sophisticated regret definition (called either swap or interval regret), when all players in a game minimize this stronger type of regret,

the empirical distribution of play is known to converge to *correlated* equilibrium [1].

Regret minimization algorithms date back to Hannan [6], and have found many applications in game theory, universal coding, and online learning. An excellent comprehensive monograph on the subject is [3].

1.3 Option Pricing

Options have been used from the time of Ancient Romans, Grecians, and Phoenicians for risk management. A call option is a contract between two parties (buyer and seller, or writer) that provides the buyer with insurance against appreciation in the price of a risky asset. It is used to hedge risk associated with financial assets (such as equities and currencies) as well as non-financial assets (commodities such as oil).

A (European) call option on a risky asset gives you the right but not the obligation to buy a risky asset on a pre-specified date, T, at a pre-specified price, K, referred to as the *strike price*. For example, a $T = 1$-year call option on IBM with a strike price $K = \$100$ gives you the right but not the obligation to buy an IBM share from the writer of the option for a price of $100. If we denote the value of the risky asset at time t by S_t then at time T the payoff of the call option is given by:

$$\max\{S_T - K, 0\}.$$

A fundamental question in finance is to determine the value of such an option today. Black and Scholes [2] studied this question in a path breaking paper that was later recognized by the 1997 Noble Prize.

We relate robust option pricing to regret minimization (following [5]). To illustrate the approach consider the following example. Suppose the current IBM share price is $100 and the risk free interest rate is zero. Assume that we have an online trading algorithm such that if we start with $100 then at time T our payoff will exceed $\max\{80, 0.8 S_T\}$, where S_T denotes IBM share price at time T. (This can be viewed as a loss of not more than 20% compared to the best asset in hindsight, namely $\max\{100, S_T\}$.) By scaling our online trading algorithm we can conclude that starting with $125 our algorithm would have a payoff that exceeds $\max\{100, S_T\}$. If we borrow $100 to initiate our online algorithm, then after paying off our (zero interest) loan, the final payoff of our online algorithm would exceeds $\max\{S_T - 100, 0\}$, which is identical to the payoff of a call option on IBM with a strike price of $100. Thus, to avoid arbitrage, the value of the call option cannot exceed the upfront cost of $25 of the online trading algorithm.

In the above example the quality of our bound is determined by the loss of our online trading algorithm relative to best static decision (which is either to buy the stock or

not to buy it). A loss of 20% translated into a bound of $25 and a better guarantee would translate into a lower loss and hence better (i.e., lower) upper bound. The work of [5] uses online regret minimization algorithms derive a bound on the option price based on the quadratic variation of the risky asset, where the regret minimization algorithms are similar in spirit to [4].

References

[1] R. J. Aumann. Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, 1:67–96, 1974.

[2] Fisher Black and Myron Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–654, 1973.

[3] Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning and Games*. Cambridge University Press, 2006.

[4] Nicolò Cesa-Bianchi, Yishay Mansour, and Gilles Stoltz. Improved second-order bounds for prediction with expert advice. In *Annual Conference on Learning Theory (COLT)*, 2005.

[5] Peter DeMarzo, Ilan Kremer, and Yishay Mansour. Online trading algorithms and robust option pricing. In *ACM Symposium on Theory of Computing (STOC)*, 2006.

[6] J. Hannan. Approximation to bayes risk in repeated plays. In M. Dresher, A. Tucker, and P. Wolfe, editors, *Contributions to the Theory of Games*, volume 3, pages 97–139. Princeton University Press, 1957.