Paper Reading Assignment 1

HU Yang

Online Convex Programming and Generalized Infinitesimal Gradient Ascent
SDSC8014 - Online learning

January 23, 2022

1 Introduction

Basically, this paper made several contributions:

1. Defined an online convex programming problem.
2. Developed generalized infinitesimal gradient ascent (GIGA) algorithm, and proved that GIGA is a very effective technique by an illustration of repeated game.
3. Proved the extensions of GIGA are universally consistent.

In the paper, the authors introduce a problem, namely *online convex programming*, where the **convex solution set is known**. However, due to its online nature, the optimization problem has to be solved iteratively. One should **select a solution point in the domain before the cost function of each step appears**. It's indeed an interesting topic with many real-world applications.

2 Technological details

In this paper, we present an algorithm for general convex functions based on gradient descent. The algorithm applies gradient descent in $\mathbb{R}^n$, and then moves back to the set of feasible points.

The algorithm has three advantages listed below.

1. Gradient descent is a simple, natural algorithm that is widely used.
2. This algorithm is capable to handle an arbitrary sequence of convex functions.
3. In online linear programs, GIGA in some circumstances perform better than an experts algorithm.

2.1 Preliminaries

We pick up some important definitions and demonstrated here:

Definition 1. *A convex programming problem consists of a convex feasible set F, and an infinite sequence $\{c^1, c^2, \dots\}$, where each $c^t : F \to \mathbb{R}$ is a convex function for each time step t. The optimal solution is the solution that* ***minimizes the cost****.*

At each time step t, an online convex programming algorithm selects a point x^t from F first, then receives the cost function c^t.(Definition 3-4 in the original paper)

Personal Remark. *The authors stated that, online algorithms do not reach solutions, but achieve certain goals instead, because Because all information is not available before decisions are made.*

Then, the paper proposes the first algorithm for online convex programming:

Algorithm 1. ***Greedy Projection*** *Select a point x^t from F and a sequence of learning rates $\eta_1, \eta_2, \dots, \mathbb{R}_+$. In time step t, after receiving a cost function, select the next vector x^{t+1} according to:*

$$x^{t+1} = P(x^t - \eta_t \nabla c^t(x^t)) \tag{1}$$

The basic idea of Eq.1 is clear: perform a gradient descent procedure based on the current point x^t, learning rate η_t, and cost function c^t. Then, if the one-step descent is out of the domain F, projection operation $P(\cdot)$ projects the result onto the nearest point of F, i.e., the point of time step $t+1$. The performance of Greedy Projection is evaluated in Section 2.1 (Regret against fixed strategy) and 2.2 (Regret against dynamic strategy).

(Ah, well, I remember such technique in **projected sub-gradient method** discussed in optimization course).

Further, the paper also proposed another similar algorithm in Section 2.3 namely *Lazy Projection* (Although the author stated it is different algorithm compared with Greedy Projection, the logic is highly identical, what's the real difference?).

2.2 Reformulation

In Section 3, the paper established a scenario of online linear programming, which is called repeated game. The repeated game contains a player and the game environment, and the game is played step by step between the player and the environment. The trajectory that records all actions played is *history*, and each history trajectory has a *utility*. Regret is a metric measuring *utility* loss between two comparable sets of actions.

The paper consider the case where $A = \{1, 2, \dots, n\}$. Before each time step in the repeated game, a distribution over actions should be selected. This can be represented as a vector in a n-standard closed simplex, the set of all points $x \in \mathbb{R}^n$ such that for all i, $x_i \geq 0$, and $\sum_{i=1}^n x_i$. A utility u instead of cost function c is proposed, then perform gradient **ascent** instead of descent. After that, the most important algorithm of the paper is presented:

Algorithm 2. ***Generalized Infinitesimal Gradient Ascent (GIGA)*** *Choose a sequence of learning rates $\{\eta_1, \eta_2, \dots\}$. Begin with an arbitrary vector $x^1 \in F$. For each round t, first play action i with probability x_i^t, then receive action $h_{t,2}$ of the environment and calculate:*

$$x^{t+1} = P(x_i^t + \eta_t u(i, h_{t,2})) \tag{2}$$

Personal Remark. *GIGA is self-oblivious, in that the strategy in the current time step can be calculated given a constant, and the past actions of the environment. Given a self-oblivious behavior σ, if for every $\epsilon > 0$ there exists a T such that, for all deterministic, oblivious environments ρ, for all $a \in A$, for all $t > T$, GIGA is* ***universally consistent*** *(See Lemma 1 in the paper and proof C).*

Here, the definition of **universally consistent** is introduced:

Definition 2. *A behavior σ is* ***universally consistent*** *if for any $\epsilon > 0$ there exists a T such that for all ρ:*

$$\Pr_{h \in F_{\sigma,\rho}} \left[\forall t > T, \frac{R(h|_t)}{t} > \epsilon \right] < \epsilon \tag{3}$$

Universally consistent means that, with high probability the average regret never surpasses ϵ again after some time.

Apart from Algorithm GIGA, a simple version of algorithm for online gradient ascent inspired by smooth fictitious play is also presented:

Algorithm 3. ***Z Fictitious Play*** *Choose $\eta > 0$. At time step t, the player does:*

$$x^t = P(\sum_{j=1}^{t-1} \eta u(i, h_{j,2})) \tag{4}$$

Section 4 discusses how to translate mixing experts algorithms $\rightarrow$ algorithms for online linear programs, and online linear programming algorithms $\rightarrow$ algorithms for online convex programs.

3 Concluding Remarks

The paper mainly defined the basics of online convex optimization, and then proposed a computationally efficient way of GIGA to solve it by performing gradient descent/ascent and re-projecting decision vector into feasible set. Given oblivious conditions, GIGA is proved to be universally consistent. Further, it attains an $O(\sqrt{T})$ regret bound proved in Theorem 1 against fixed strategy.