POLITECNICO
MILANO 1863

Project for the course
“Advanced Programming for Scientific Computing”

Tutorized by Nicola Gatti

MARKOV PERSUASION PROCESSES: LEARNING TO PERSUADE FROM SCRATCH

Miguel Alcañiz Moya

Lecturers: Prof. L. Formaggia and Prof. C. De Falco

Date: July 17th, 2024

Contents

1 Introduction **1**

1.1 Introduction to MPPs 1

1.2 Bayesian Persuasion 2

1.3 Markov Persuasion Processes 3

1.4 Optimisitc Persuasive Policy Search 5

2 Implementation **6**

2.1 Utilities 6

2.2 Episode Generator 6

2.2.1 Prior function 6

2.2.2 Transition function 7

2.2.3 Rewards container 7

2.3 Opt Opt 7

2.4 Markov Persuasion Process 7

3 Test **8**

3.1 Example 8

Abstract

In this project we try to implement the algorithms from the paper Markov Persuasion Processes: Learning to Persuade from Scratch [1]. In Bayesian persuasion, an informed sender strategically discloses information to a receiver so as to persuade them to undertake desirable actions. Recently, Markov persuasion processes (MPPs) have been introduced to capture sequential scenarios where a sender faces a stream of myopic receivers in a Markovian environment. The MPPs studied so far in the literature suffer from issues that prevent them from being fully operational in practice, e.g., they assume that the sender knows receivers' rewards. We fix such issues by addressing MPPs where the sender has no knowledge about the environment. We are testing a learning algorithm for the sender in which attain regret sublinear in the number of episodes T while being persuasive.

Chapter 1

Introduction

1.1 Introduction to MPPs

Bayesian persuasion studies how an informed sender should strategically disclose information to influence the behavior of an interested receiver. The vast majority of works on Bayesian persuasion focuses on one-shot interactions, where information disclosure is performed in a single step. Despite the fact that real-world problems are usually sequential, there are only few exceptions that consider multi-step information disclosure [4].

In particular, Wu et al. (2022) ([4]) initiated the study of Markov persuasion processes (MPPs), which model scenarios where a sender sequentially faces a stream of myopic receivers in an unknown Markovian environment. In each state of the environment, the sender privately observes some information—encoded in an outcome stochastically determined according to a prior distribution—and faces a new receiver, who is then called to take an action. The outcome and receiver's action jointly determine agents' rewards and the next state. In an MPP, sender's goal is to disclose information at each state so as to persuade the receivers to take actions that maximize long-term sender's expected rewards. MPPs find application in several real-world settings, such as e-commerce and recommendation systems. For example, an MPP can model the problem faced by an online streaming platform recommending movies to its users. Indeed, the platform has an informational advantage over users (e.g., it has access to views statistics), and it exploits available information to induce users to watch suggested movies.

Nevertheless, the MPPs studied by Wu et al. (2022) ([4]) suffer from several issues that prevent them from being fully operational in practice. In particular, they make the rather strong assumption that the sender has perfect knowledge of receiver's rewards. This is unreasonable in real-world applications. For instance, in the online streaming platform example described above, such an assumption requires that the platform knows everything about users' (private) preferences over movies.

In our setting we considerably relax the assumptions of Wu et al. (2022) ([4]), by addressing MPPs where the sender does not know anything about the environment. We consider settings in which the sender has no knowledge about transitions, prior distributions over outcomes, sender's stochastic rewards, and receivers' ones. Thus, they have to learn all these quantities simultaneously by repeatedly interacting with the MPP.

Summaring, the algorithm implemented is mixing the classic Bayesian Persuasion problem with a Markov decision problem, trying to bring this area to a more realistic setting for the real world in which we do not know how does the enviroment work, so we have to learn about it by interacting with receivers. The enviroment unkown by the sender refers to the rewards won from the interactions and the probability transitions between the states.

1.2 Bayesian Persuasion

The classical Bayesian persuasion framework introduced by Kamenica and Gentzkow (2011) [1] models a one-shot interaction between a sender and a receiver. The latter has to take an action a from a finite set A, while the former privately observes an outcome ω sampled from a finite set Ω according to a prior distribution $\mu \in \Delta(\Omega)$, which is known to both the sender and the receiver. The rewards of both agents depend on the receiver's action and the realized outcome, as defined by the functions

$$r_S, r_R : \Omega \times A \to [0, 1],$$

where $r_R(\omega, a)$ and $r_S(\omega, a)$ denote the rewards of the sender and the receiver, respectively, when the outcome is $\omega \in \Omega$ and action $a \in A$ is played.

The sender can strategically disclose information about the outcome to the receiver, by publicly committing to a signaling scheme ϕ, which is a randomized mapping from outcomes to signals being sent to the receiver. Formally,

$$\phi : \Omega \to \Delta(S),$$

where S denotes a suitable finite set of signals. For ease of notation, we let $\phi(\cdot|\omega) \in \Delta(S)$ be the probability distribution over signals employed by the sender when the realized outcome is $\omega \in \Omega$, with $\phi(s|\omega)$ being the probability of sending signal $s \in S$.

The sender-receiver interaction goes on as follows:

1. the sender publicly commits to a signaling scheme ϕ;

2. the sender observes the realized outcome $\omega \sim \mu$ and draws a signal $s \sim \phi(\cdot|\omega)$; and

3. the receiver observes the signal s and plays an action. Specifically, after observing s under a signaling scheme ϕ, the receiver infers a posterior distribution over outcomes and plays a best-response action $b_\phi(s) \in A$ according to such distribution. Formally:

$$b_{\phi}(s) \in \arg\max_{a \in A} \sum_{\omega \in \Omega} \mu(\omega)\phi(s|\omega) r_R(\omega, a),$$

Where the expression being maximized encodes the (unnormalized) expected reward of the receiver. As it is customary in the literature [1], we assume that the receiver breaks ties in favor of the sender, by selecting a best response maximizing sender's expected reward when multiple best responses are available.

The goal of the sender is to commit to a signaling scheme ϕ that maximizes their expected reward, which is computed as follows:

$$\sum_{\omega \in \Omega} \mu(\omega) \sum_{s \in S} \phi(s|\omega) r_S(\omega, b_{\phi}(s)).$$

If you need a better understanding of the problem have a look at the paper *Public Signaling in Bayesian Ad Auctions* [2].

1.3 Markov Persuasion Processes

A Markov persuasion process (MPP) (Wu et al., 2022) [4] generalizes the one-shot Bayesian persuasion framework by Kamenica and Gentzkow (2011) [1] to settings in which the sender sequentially interacts with multiple receivers in a Markov decision process (MDP). In an MPP, the sender faces a stream of myopic receivers who take actions by only accounting for their immediate rewards, thus disregarding future ones.

A Markov persuasion process (MPP) (Wu et al., 2022) generalizes the one-shot Bayesian persuasion framework by Kamenica and Gentzkow (2011) to settings in which the sender sequentially interacts with multiple receivers in a Markov decision process (MDP). In an MPP, the sender faces a stream of myopic receivers who take actions by only accounting for their immediate rewards, thus disregarding future ones.

Formally, an (episodic) MPP is defined by means of a tuple

$$M = (X, A, \Omega, \mu, P, \{r_{S,t}\}_{t=1}^{T}, \{r_{R,t}\}_{t=1}^{T}),$$

where:

- T is the number of episodes.
- X, A, and Ω are finite sets of states, actions, and outcomes, respectively.
- $\mu : X \to \Delta(\Omega)$ is a prior function defining a probability distribution over outcomes at each state. For ease of notation, we let $\mu(\omega \mid x)$ be the probability with which outcome $\omega \in \Omega$ is sampled in state $x \in X$.
- $P : X \times A \to \Delta(X)$ is a transition function. For ease of notation, we let $P(x' \mid x, a)$ be the probability of moving from $x \in X$ to $x' \in X$ by taking action $a \in A$.

- $\{r_{S,t}\}_{t=1}^{T}$ is a sequence specifying a sender's reward function $r_{S,t} : X \times \Omega \times A \to [0,1]$ at each episode t. Given $x \in X, \omega \in \Omega$, and $a \in A$, each $r_{S,t}(x,\omega,a)$ for $t \in [T]$ is sampled independently from a distribution $\nu_S(x,\omega,a) \in \Delta([0,1])$ with mean $r_S(x,\omega,a)$.

- $\{r_{R,t}\}_{t=1}^{T}$ is a sequence defining a receivers' reward function $r_{R,t} : X \times \Omega \times A \to [0,1]$ at each episode t. Given $x \in X, \omega \in \Omega$, and $a \in A$, each $r_{R,t}(x,\omega,a)$ for $t \in [T]$ is sampled independently from a distribution $\nu_R(x,\omega,a) \in \Delta([0,1])$ with mean $r_R(x,\omega,a)$.

For ease of presentation, we focus w.l.o.g. on episodic MPPs enjoying the loop-free property, as customary in the online learning in MDPs literature. In a loop-free MPP, states are partitioned into $L+1$ layers $X_0, \ldots, X_L$ such that $X_0 := \{x_0\}$ and $X_L := \{x_L\}$, with x_0 being the initial state of the episode and x_L being the final one, in which the episode ends. Moreover, by letting $\mathcal{K} := [0 \ldots L-1]$ for ease of notation, $P\left(x' \mid x, a\right) > 0$ only when $x' \in X_{k+1}$ and $x \in X_k$ for some $k \in \mathcal{K}$

At each episode of an episodic MPP, the sender publicly commits to a signaling policy $\phi : X \times \Omega \to \Delta(\mathcal{S})$, which defines a probability distribution over a finite set $\mathcal{S}$ of signals for the receivers for every state $x \in X$ and outcome $\omega \in \Omega$. In our setting we will take $\Omega = A$. For ease of notation, we denote by $\phi(\cdot \mid x, \omega) \in \Delta(\mathcal{S})$ such probability distributions, with $\phi(s \mid x, \omega)$ being the probability of sending a signal $s \in \mathcal{S}$ in state x when the realized outcome is ω.

As customary in Bayesian persuasion settings, a revelation-principle-style argument allows to focus w.l.o.g. on signaling policies that are direct and persuasive. Formally, a signaling policy is direct if the set of signals coincides with the set of actions, namely $\mathcal{S} = A$. Intuitively, in such a case, signals should be interpreted as action recommendations for the receivers.

So our first algorithm to we implement is the sender-receivers interaction in the sequential decision process defined previously.

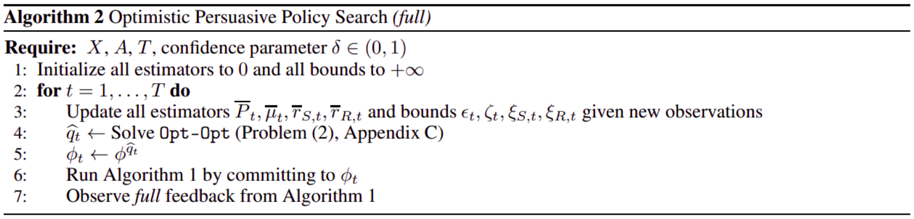
Algorithm 1 Sender-Receivers Interaction at $t \in [T]$

```
All the rewards $r_{S,t}(x,\omega,a), r_{R,t}(x,\omega,a)$ are sampled
Sender publicly commits to $\phi_t : X \times \Omega \to \Delta(A)$
The state of the MPP is initialized to $x_0$
for $k = 0, \ldots, L-1$ do
    Sender observes outcome $\omega_k \sim \mu(x_k)$
    Sender draws recommendation $a_k \sim \phi(\cdot|x_k, \omega_k)$
    A new Receiver observes $a_k$ and plays it
    The MPP evolves to state $x_{k+1} \sim P(\cdot|x_k, \omega_k, a_k)$
    Sender observes the next state $x_{k+1}$
Sender observes feedback for every $k \in [0 \ldots L-1]$:
    • full $\to r_{S,t}(x_k, \omega_k, a), r_{R,t}(x_k, \omega_k, a)\ \ \forall a \in A$
    • partial $\to r_{S,t}(x_k, \omega_k, a_k), r_{R,t}(x_k, \omega_k, a_k)$
```

The interaction between the sender and the stream of receivers at episode $t \in [T]$ is described in Algorithm 1. Let us remark that sender and receivers do not know anything about the transition function P, the prior function mu, and the rewards $r_{R,t}(x,\omega,a)$, $r_{R,t}(x,\omega,a)$ (including their distributions). At the end of each episode, the sender gets to know the triplets (x_k, ω_k, a_k) $\forall k \in K$ that are visited during the episode, and an additional feedback about rewards. Although in the paper this project it's inspired of they consider two types of feedback, we just consider the feedback of the rewards for all the triplets (x_k, ω_k, a_k) visited during the episode.

1.4 Optimisitc Persuasive Policy Search

The algorithm proposed in [1] we propose an algorithm called Optimisitc Persuasive Policy Search (OPPS). At each episode, the algorithm solves a variation of the offline optimization problem called Opt-Opt, defined in Appendix C of [1]. Crucially, by using occupancy measures, Opt-Opt can be formulated as a linear program, and, thus, it can be solved efficiently. Although in this project we couldn't finally make it to program this part. So the signaling scheme obtained out of this problem in our program is just a random function which outputs a valid signaling scheme. Because of this issue, significant results proved in [1] could not be tested. The Optimistic Persuasive Policy Search with full feedback of the following code correspond to the incomplete OPPS algorithm programmed in this project.



```
Algorithm 2 Optimistic Persuasive Policy Search (full)
Require: $X$, $A$, $T$, confidence parameter $\delta \in (0, 1)$
Initialize all estimators to 0 and all bounds to $+\infty$
for $t = 1, \ldots, T$ do
    Update all estimators $\overline{P}_t, \overline{\mu}_t, \overline{r}_{S,t}, \overline{r}_{R,t}$ and bounds $\epsilon_t, \zeta_t, \xi_{S,t}, \xi_{R,t}$ given new observations
    $\widehat{q}_t \leftarrow$ Solve Opt-Opt (Problem (2), Appendix C)
    $\phi_t \leftarrow \phi^{\widehat{q}_t}$
    Run Algorithm 1 by committing to $\phi_t$
    Observe full feedback from Algorithm 1
```

For more understanding of the algorithm and the analysis of its performance look at the paper 'Markov persuasion processes: learning to persuade from scratch' [1].

Chapter 2

Implementation

The algorithm implementation is composed by a c++ code formed by three main parts, the episode generator files, the Opt Opt files and the Markov Persuasion Process part. Each part consists of a *.hpp* file and a *.cpp* file. All these files are kept in the */src* folder.

2.1 Utilities

Describe the includes, the alias and the declaration of the template classes.

2.2 Episode Generator

This first part is where the environment setting for the Markov Persuasion Process is defined. Referring to this environment setting the following functions and sets:

- The prior function $\mu : X \to \Delta(\Omega)$. Defining a probability distribution over outcomes at each state.
- The transition function $P : X \times A \to \Delta(X)$. Defining the probability to move to another possible next state given a chosen action.
- The rewards container. Which is used to save the rewards for the sender and receiver for every possible state, action and outcome.

2.2.1 Prior function

The header file defines the `prior` class. Its protected components are the following:

- The protected member function *size_t* L correspond to the number of partitions of the states set $\mathcal{S}$.

- The protected member function *size_t A* correspond to the number of possible actions of the set $\mathcal{A}$.
- The protected member function *TensorI states* correspond to the vector of states of every partition of $\mathcal{S}$.
- The protected member function *Tensor3D priorD* correspond to a three dimensional tensor of doubles that keeps the probability of having an outcome o in each state s of partition l ($priorD[l][s][o]$).

```
    protected:
        size_t L;
        size_t A;
        TensorI states;
        Tensor3D priorD;
    };
```

2.2.2 Transition function

2.2.3 Rewards container

2.3 Opt Opt

2.4 Markov Persuasion Process

Chapter 3

Test

3.1 Example

Escribe aquí las conclusiones de tu proyecto.

Bibliography

[1] Francesco Bacchiocchi et al. *Markov Persuasion Processes: Learning to Persuade from Scratch.* 2024. arXiv: `2402.03077 [cs.GT]`.

[2] Francesco Bacchiocchi et al. *Public Signaling in Bayesian Ad Auctions.* 2022. arXiv: `2201.09728 [cs.GT]`. URL: `https://arxiv.org/abs/2201.09728`.

[3] Emir Kamenica and Matthew Gentzkow. "Bayesian Persuasion". In: *American Economic Review* 101.6 (2011).

[4] Jibang Wu et al. *Sequential Information Design: Markov Persuasion Process and Its Efficient Reinforcement Learning.* 2022.