Online Learning Applications

Project Requirements

Recap on evaluation

The evaluation is composed by two parts:

Written exam

- Scheduled on the official calls (see Online Services)
- max 16 points
- Questions about theory

Project

- Three scheduled project presentations (see last slides)
- max 16 points
- Groups of at most 4-5 students

Recap on evaluation

The evaluation is composed by two parts:

Written exam

- Scheduled on the official calls (see Online Services)
- max 16 points
- Questions about theory

Project

- Three scheduled project presentations (see last slides)
- max 16 points
- Groups of at most 4-5 students ⚠ **If you can't find a group, send me an email by Friday (18/04) and I will match you with other students without a group.**
- The goal is to develop algorithms for a complex problem
- Includes modeling and coding

Overview of the project

Goal

The goal of the project is to design online learning algorithms to sell **multiple** types of products under **production constraints**.

Setting

A company has to choose prices dynamically.

Parameters

- Number of rounds T
- Number of types of products N
- Set of possible prices P **(small and discrete set)**
- Production capacity B ⚠ **For simplicity, there is a total number of products B that the company can produce (independently from the specific type of product)**

Buyer behavior

- Has a valuation v_i for each type of product in N
- Buys all products priced below their respective valuations

Interaction

At each round $t \in T$:

1. The company chooses which types of product to sell and set price p_i for each type of product
2. A buyer with a valuation for each type of product arrives
3. The buyer buys a unit of each product with price smaller than the product valuation

Requirement 1: Single product and stochastic environment

Environment

Build a **stochastic** environment:

- A distribution over the valuations of a single type of product

Design algorithms for a single product and stochastic environment

Algorithm

Build a pricing strategy using UCB1 **ignoring the inventory constraint**.

Algorithm

Build a pricing strategy extending UCB1 to handle the **inventory constraint**.

Design algorithms for a single product and stochastic environment

Algorithm

Build a pricing strategy using UCB1 **ignoring the inventory constraint**.

Algorithm

Build a pricing strategy extending UCB1 to handle the **inventory constraint**.

Hint

Extend the "UCB-like" approach that we saw for auctions to the pricing problem.

Requirement 2: Multiple products and stochastic environment

Environment

Build a **stochastic** environment:

- A joint distribution over the valuations of all the types of products

Design algorithms for multiple products and stochastic environment

Algorithm

Build a pricing strategy using Combinatorial-UCB **with the inventory constraint**.

Design algorithms for multiple products and stochastic environment

Algorithm

Build a pricing strategy using Combinatorial-UCB **with the inventory constraint**.

Hint

Extend the "UCB-like" approach that we saw for auctions to the combinatorial pricing problem.

Requirement 3: Best-of-both-worlds algorithms with a single product

Environment

Use the **stochastic** environment already designed:

- A distribution over the valuations of a single product

Build a **highly non-stationary** environment. At a high level, it should include:

- A sequence of valuations of the product (e.g., sampled from a distribution that changes **quickly** over time)

Design best-of-both-worlds algorithms with a single product

Algorithm

Build a pricing strategy using a primal-dual method **with the inventory constraint**.

Design best-of-both-worlds algorithms with a single product

Algorithm

Build a pricing strategy using a primal-dual method **with the inventory constraint**.

Hint

To design a primal-dual method, extend the results on "general auctions" to f and c suitable for the pricing problem.

Requirement 4: Best-of-both-worlds with multiple products

Environment

Use the **stochastic** environment already designed:

- A joint distribution over the valuations of all the types of products

Build a **highly non-stationary** environment. At a high level, it should include:

- A sequence of correlated valuations for each type of product (e.g., sampled from a distribution that changes **quickly** over time)

Design best-of-both-worlds algorithms with multiple products

Algorithm

Build a pricing strategy using a primal-dual method **with the inventory constraint**.

Design best-of-both-worlds algorithms with multiple products

Algorithm

Build a pricing strategy using a primal-dual method **with the inventory constraint**.

Hint

To design a primal regret minimizer, notice that the pricing problem "decomposes". It is sufficient to design an (adversarial) regret minimizer for each type of product.

Requirement 5: Slightly non-stationary environments with multiple products

Non-stationary environment

Build a **slightly non-stationary** environment for the pricing problem. At a high level:

- Rounds are partitioned in intervals
- In each interval the distribution of products valuations is fixed
- Each interval has a different distribution

Algorithm

Extend Combinatorial-UCB with **sliding window**

Compare

Compare the performance of:

- Combinatorial-UCB with sliding window
- The primal-dual method

Deliverable and presentation

Deliverable

Before the presentation you should deliver:

- Some slides describing the project and the results in detail:
  - **High level details** about the implemented algorithms
  - **graphs** of the empirical results
- A link to a GitHub repository with the code of the project

The presentation should be conducted as follows:

- You will have 20 minutes to deliver your presentation: additional 10 minutes will be used for questions and answers
- A detailed discussion of unexpected results is appreciated (e.g. the algorithm does not achieve a sublinear regret)

Details about the project presentation schedule

You have three possible sessions to present the project in **July, September, December**;

- The deadline for the July project submission is **11th of July**
- The presentations will be in the days following the submission