Dynamic Reserve in Second-price Auctions

Online learning methods for sellers and buyers

by

Lorenzo Croissant

under the supervision of

Dr. Marc Abeille

and the tutorage of

Dr. Vianney Perchet

Internship Report submitted in partial fulfilment of the

degree of *Master in Mathematics*

M2 Mathématiques, Vision, Apprentissage

Thursday 12th of September 2019

Dedication

I would like to extend thanks to all those who have supported me during this academic year and in particular the last six months as a part of Criteo. I thank all my team members and the support staff for their warm welcome. I would like to thank the auction theory team for their help on the auction theoretic side of this project: Vianney for his high level guidance and most especially Thomas and Clement for their continuous collaboration. I am very indebted to Marc for his daily assistance but especially for his long-term vision and holistic approach to solving reinforcement learning. I am particularly grateful for his help in organising a Ph.D. candidacy and his excellent suggestions of supervisors. I would like to also extend my gratitude to the close friends which have supported me and pushed me forwards during the year. I am very grateful for the wonderful cooperation I enjoyed on various projects during the year with Solange, Bastien, Thibault and Baptiste to name only a few.

Contents

Chapter 1

Introduction

The internship which this report is about was conducted at Criteo's AI Lab, under the supervision of Marc Abeille and Vianney Perchet and the topic of research was to investigate reinforcement learning methods for repeated second price auctions with dynamic reserve prices. This chapter will attempt to make the previous sentence clear for all readers. We will begin by some general context about the online advertising market and Criteo, before moving on to the theoretical context, first auction theory, then reinforcement learning. The objective is to highlight the speci ficities and the difficulties of the problem, and thus to get a general grasp on it so that we can begin solving it.

1.1 The online advertising ecosystem

Criteo is a French company, founded in 2005, which focuses on the market of online advertising. This sector of industry is omnipresent and yet remains mostly unknown of the general public. This section will benefit not only the mathematical research by highlighting some of the particular difficulties that inspired our line of work, but it will benefit the reader's general knowledge since online advertising is a corner stone of the internet we have become accustomed to over the last decade. We focus first on the structure of the market itself, and then we will succinctly present Criteo itself and the environment of the work.

1.1.1 The actors of the online advertising market

The types of agents in the online advertising market is quite similar to the market for old fashioned advertising, such as billboards or TV ads. On one end, someone has a pool of users to which they can display ads, and display space to sell. On the other end, companies have products to sell and are ready to pay money to advertise them. In the middle there will be one or two middle men, such as marketing agencies. The middle men make the exchange happen and the money change hands. What actually sets the two markets apart on a qualitative level is the type of advertising.

In a traditional TV ad you can target a pool of users which is the pool of users who are watching the TV program in which the ad is embedded. So, you can advertise beer during a football match, and it is likely to be quite effective, but your ad will be wasted on a large share of the audience, who will forget, or do not drink beer. It is an unavoidable fact of the medium that neither TV ads nor billboards will ever be accurately targeted, which means the value generated per viewer of the ad is minimal. In online advertising it is entirely possible to have ads who will generate a purchase on nearly every display, making them vastly more effective and thus valuable.

Targeted advertising leveraging user's data is the bread and butter of online advertising. Cookies collect behavioural information about you and your preferences. Data for millions or even billions of users are aggregated and analysed using large scale learning methods which allow specialised companies like Criteo to very accurately predict the effectiveness of an ad. In parallel, automation of the process of selling advertising space at very high frequency makes it possible to actually display ads to individual people, turning this accurate appraisal of users into actual sales.

Now, the market functions as such: on the supply side, individual website owners, *publishers*, stream the data on their visitors and offer up display space. They most often will contract with an ad-exchange (a.k.a. a marketplace) to do this, who will handle all the automated process for publishers in exchange for a small commission. The ad-exchange will then organise an auction to sell the display space to the highest bidder, and, when it is sold, will pay the publisher their share and send them the ad to display. All this happens in the time the web-page takes to load, in roughly 15 milliseconds.

Market places allow publishers without the know-how and infrastructure to construct efficient payment mechanisms to fructify their pages; in a symmetric manner, Criteo allows companies wishing to advertise online, but without large technical expertise, to benefit from Criteo's internal data processing systems and get more revenue than by bidding directly in the ad auctions themselves. For the main part the competitive advantage is that Criteo has large volumes of internal data, specialised engineers, and dedicated systems which optimise the process in a way that companies could not do individually. One way to view this market is as two collusion rings (marketplaces and companies like Criteo) on either side of supply and demand interacting through repeated auctions.

The particularity that will be important in this paper is that these repeated auctions create a lot of data, which is available for both marketplaces and bidders to try and twist the auctions to increase their revenue. On the one hand, revenue can not be made by both players in this repeated game, and the competition does genuinely revolve around nipping into the opponents margin. But it is very important to understand that on the other hand, it is not a repeated zero-sum game. In an auction, no one will make any money at all if the item is not sold, so all players must cooperate somewhat, and it is only when they are cooperating enough and sales take place that they can compete over the price of the sale.

1.1.2 Presentation of Criteo AI Lab

Criteo is a fourteen year old French leader in ad-tech which is listed in New York and does business with companies all across the world. Unlike Google and Facebook, it is independent of other internet services and offers clients the possibility to target various marketing campaigns at internet users. The core business is retargeting (bringing back shoppers who have begun but not completed a purchase) but they offer many different kinds of services which extend to brand recognition or generation of traffic to brick and mortar stores.

The business model of Criteo isn't, however, what we're particularly interested in. It is its dual position as both an auctioneer and a bidder in ad-exchanges. In the online advertising market Criteo acts as a bidder when it buys ad-space from ad-exchanges but it also has to act as an auctioneer when determining which of its clients to support in bids. It is therefore in the unique position of profiting from improvements on both sides of the auction.

To maintain a competitive advantage, Criteo has an in house research team, consisting of applied and also theoretical researchers. In the theoretical part, the Explore-Exploit Learning team studies problems around reinforcement learning, online and stochastic optimisation, control theory along with auction theory both game theoretical and computational. It is in this team that the project was conducted.

1.2 The repeated auction problem

As we highlighted in the previous section, a particularity of the online advertising market, which prompted the line of work to which this report contributes, is the dynamic nature of the repeated auctions. There are over 10 billion auctions a day that Criteo partakes in, and most of them are directly with Google. For ad-exchanges, the way to maximise the revenue is to tweak the mechanism of the auction repeatedly to incrementally increase revenue, while for the bidder it is to falsify their bid distribution to push the seller to lower the price. Of course, when one finds a new trick to increase their revenue the other one will adapt to it by coming up with a trick of their own, etc.

1.2.1 From the seller's perspective

Let's begin our analysis of this general repeated auction problem with the seller's perspective. We need to talk somewhat about the seller first in order to better understand the bidder later. For a seller, the repeated auction, as we saw above, involves tuning the

mechanism to increase their revenue. In the real world most auctions are second price with reserve (Vickrey auctions) and this is still the main format in online advertising too.

For some time now, Google has been shifting their auctions to first price to improve their revenue, but we will ignore this development, choosing to only consider second price with reserve as the attribution and payment rules of the mechanism. In this, we are essentially placing constraints on what the seller can do, but it will allow us to convert the problem of optimisation from mechanisms to only reserve prices, which are a much simpler class of objects to deal with. And simplicity is in fact key to these very large scale problems: the main roadblock for Google in the optimisation of their auction is that the algorithm they run to improve their reserve prices takes a couple days to update every time it needs to be recomputed. With the frequency of auctions we are dealing with, any small speed improvement matters.

We consider lazy second price auctions implicitly throughout this report, and we do not concern ourselves with ties, but we will allow for reserve prices to be personalised to each bidder. In this type of auction, before bids are placed, the seller picks a reserve price per bidder, the bids are the tallied and the highest one is compared to the appropriate reserve price and if it is cleared the payment is the maximum of the reserve and the second highest bid. It is a standard auction format for second price with reserve, and we preferred it to eager auctions where the reserve has to be cleared before bids are tallied and the highest clear bid gets the item.

In an eager second price auction the revenue is generally higher than in the lazy auction. However, the computation of the optimal price is much harder: it is NP-Hard in fact[33]. However, it has been shown, again by Paes Leme et al. that the *monopoly price*, which is the optimal price with only one buyer, is the optimal price in lazy second price auctions and that it is a 2 approximation to the optimal price in the eager version[33]. For F the CDF of the bid distribution $\mathcal{B}$, the monopoly price of the bid distribution is defined as

$$\max_r r(1 - F(r)) .$$

Further, Roughgarden and Wang showed that an eager auction with monopoly is a 2-approximation to the overall optimal auction of Myerson[29]. Thus, the problem of learning the monopoly price of each bidder from a sequence of lazy second price auctions is very valuable, and profits the seller well beyond this format.

Since the monopoly price defines the optimal price in a second price auction with only one buyer (which is known as a *posted price* auction). We can consider the competition absent, thanks to our individualised reserve prices, which leads to the following problem formulation. At time t the seller picks reserve prices r, the bidder then computes a value v_t for the item. The bidder has a strategy β and bids $b_t = \beta_i(v_t)$ which the seller will model by assuming that $b_t \sim \mathcal{B}$. Per the rules of lazy second price auctions, $b_t \geq r_t$ then the buyer wins the auction and pays $b_t \vee r_t$.

In a single second-price auction, it is optimal for buyers to bid their valuation $b_t = v_t$. But in the repeated case it is unlikely that this result still holds. There are several lines of work that approach this new problem from different sets of hypotheses.

First line of research is the very conservative one which would consider bidders are truthful, *i.e.* they bid their valuation: $b_t = v_t$ like in the one-shot auction. We label it as conservative in the sense that it supposes that the optimal behaviour for the static one-shot problem is not too sub-optimal in the dynamic one, so that bidders retain the same behaviour. This makes the problem much simpler as it is simply a matter of learning the monopoly price from a sample of iid values (bids).

The problem with the first approach is that it makes an assumption which we know to be false. Being truthful in many repeated auctions can be so sub-optimal, it is wishful thinking. The second line of research is to assume that bidders are somewhat strategic but that, due to some internal constraints, they are not entirely unconstrained. This assumption has the most empirical basis: as we stated previously auctions are not zero-sum games, bidders are incentivised to partake in the auction if they value the item. The practical consequence of this is that companies that bid in marketplaces have to win auctions and make some profit to stay afloat. For an interesting strategy exploiting this constraint on buyer see Amin et al.[1].

In the third line of research, bandit theorists have focused on the problem under adversarial assumptions, which are mathematically motivated by our ignorance of which strategies the buyer is considering in order to maximise his revenue, but which are vastly more difficult than the real problem. We know that *real* bidders can not be completely adversarial, since submitting highly volatile (*e.g.* random) bids would probably lead them straight to bankruptcy. The optimisation considered in these works is not the most relevant to the practical repeated auction problem, whose particularity is competition between a seller and buyer for profit but without being a zero-sum game.

In this report we will focus on the first line of research in revenue maximisation, but in chapter 3 we will demonstrate a method which can also cover the second direction somewhat by adapting to non-stationary bid distributions. The question is now, how do we maximise $r(1 - F(r))$ when: F is unknown, F may change somewhat over time. These are the object of discussion in chapter 3, we have spoken so far only about the high level considerations required to establish the model. We needed to draw the problem with some broad strokes now so that we could provide the context in which to analyse bidder learning. This was the original purpose of the internship wherefrom we deviated so as to more deeply explore the seller problem which repeatedly eluded a complete understanding.

1.2.2 From the bidder's perspective

The revenue maximisation problem from the bidder is similar but not entirely symmetric to the seller's problem. This is a consequence of the intermediate status of auctions as being neither zero-sum nor collaborative games but a game where a minimum of cooperation is required but the amount of cooperation beyond the minimum leads to competition over the margin of the buyer. Recall that the bidder i draws at time t a value $v_{i,t}$ and his revenue from the auction is 0 if he loses and the difference between $v_{i,t}$ and his payment if he wins. In terms of his reserve $r_{i,t}$ and the bids submitted by the competition $\bar{b}_{i,t}$ in round t this profit U is given by

$$U_{i,t}(b_t) := (v_{i,t} - \bar{b}_{i,t} \vee r_{i,t})\mathbb{I}\{v_{i,t} \geq \bar{b}_{i,t} \vee r_{i,t}\}\,.$$

In a second price auction with reserve, it is Pareto dominant to bid $b_i = v_i$ (truthfully), so this is how a bidder would maximise his expected revenue in an auction. When considering repeated auctions, it matters whether or not the seller is oblivious to the buyer's bids. For example if the seller is repeatedly selling a non-perishable item and sets a reserve which represents the resale value of the item then there is no reason for the seller to adjust his reserve based on the bids. This can happen for example if a seller is trying to liquidate a stock of items but is not under the pressure of time and prefers a relatively risk-averse and low-effort sale strategy. If the seller is oblivious it is obviously also dominant to always bid truthfully.

However, in a repeated auction where the seller is adaptive it becomes very suboptimal for the buyer to stay truthful. If they do, the seller could learn their bid distribution, and thus their appraisal ϕ_i so that he can set his reserve price as $r_i = v_i - \epsilon$ for some ϵ to perform *full surplus extraction*. This is the worst case scenario of a repeated auction, where it is still weakly dominant to bid truthfully, but the expected revenue of the buyer can be made arbitrarily small by the seller, leaving him with no profit. This is what a buyer wants to avoid at all cost.

With the frequency of auctions that are involved in the online advertising market, it is logical to try and maximise the asymptotic average payoff of the buyer, which must be positive and whose worst value (0) is attained by being truthful as we just saw. We define the objective as a function of collected rewards as such:

$$\rho = \mathbb{E}\left[\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} U_t(b_t)\right]\,.$$

This quantity depends on two parameters which are hidden in the expectation: the policy π, and the environment's dynamic P.

The environment's dynamic represents the (presumably stochastic) evolution of the seller's behaviour at each time t. In our case[1] it represents the distribution of $r_{i,t+1}$ given $r_{i,t}$

[1]Stochastic, Markovian and stationary update of reserves.

and $\boldsymbol{b}_t$: $r_{i,t+1} \sim P(r_{i,t}, \boldsymbol{b}_t)$. The policy term is the sequence of functions which determine what the agent should bid at time t: $\pi = \{\pi_1, \pi_2, \dots\}$ with $b_{i,t} = \pi_t(r_{i,t-1}, v_{i,t})$. Designing an optimal strategy then is equivalent on a high level to designing an algorithm to solve

$$\max_{\pi} \rho(\pi) = \mathbb{E}_{P,\mathcal{X},\mathcal{B}_{-i}} \left[\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} U_{i,t}(\pi_t(r_{i,t-1}, v_{i,t})) \right] . \tag{1.1}$$

Solving problem 1.1 is not obvious, and the study of the existence of solutions alone would be a significant endeavour. Rather than study these questions, it is customary in RL to propose an algorithm, and study whether it arrives at the optimum asymptotically, and study its speed of convergence. This speed of convergence is measured as a *regret*, which we can also consider as a form of quantifier for the finite time efficiency of the algorithm. In general this regret is the difference between the reward collected by a *comparator* and the algorithm.

1.3 Reinforcement learning

Reinforcement learning (RL) is the branch of machine learning which deals with learning from interactions with a task rather than from examples of the task. As a very simple model consider an *agent* interacting sequentially with an *environment*, where each interaction grants some *reward* to the agent indicating how well it is performing at the given task. Learning the task then becomes maximising the collected reward. There are many different formulations of RL problems which will give rise to particular sub-problems, but they all share some form of *learning* (understanding the environment) and *planning* (deciding which sequences of actions generate high reward).

1.3.1 A survey of different methodologies in RL

In this section we are going to review the different forms of RL that exist on a very high level. Most of these are probably going to be familiar to the reader so we will not dwell on them. The objective of this section is to understand what each assumption in an RL problem formulation will imply in terms of sub-tasks to solve. This will ensure we start out with a good understanding of RL and will allow us to determine where the difficulties are when formulating the problem we focus on in this report in section 1.3.2.

The first axis of discussion is over the information structure of the environment. This spectrum ranges from the easiest problems, *full information*, where the agent gets data about the reward that could have been generated by any action, to the impossible problem of no information at all. In full information, the problem of learning is quite easy since

all actions provide an equal amount of information; learning in full information is just estimation. This leaves full information problems as essentially planning problems and they are generally thus treated as sub-fields of optimisation. Between full and absent information lie all partial information problems, which make up most real world problems of RL. Information can be partial for many reasons: classical statistics problems like censoring, or pure interaction problems where only the outcome of the taken action is observed, or still because the environment may have reactions conditional on past actions.

Regardless of the cause for information loss, most if not all partial information problems require the agent to perform exploratory actions, whose goal is to gather information, not only actions which the agent thinks will maximise its reward. This naturally gives rise to a trade-off: either exploit your partial knowledge now and potentially be mistaken, or explore sub-optimal actions to be sure they are correctly assessed as sub-optimal. This trade-off between exploration and exploitation is the hardest part of reinforcement learning. It is easily addressed in simple cases like bandits, but is holding back the resolution of more complex settings in the field.

Next we would identify the distinction in the types of environment dynamics that our system considers. By this we mean the type of evolution the environment can have in response to time or the actions of the agent. If the environment is completely stationary we recover the classical bandit problem, while an example of an environment with simple dynamics would be Markov Decision Processes (MDP) with multiple states. In an MDP the environment has an internal state which is updated according to a stationary Markov chain conditional to the current state. MDPs are the common object of study since they generalise bandits (bandits are single state discrete MDPs) to many problems, while retaining learnable structure. In contrast, if there are no assumptions on the evolution of the system one is under adversarial reward assumptions which have no structure and constitute worst-case problems which are much harder to learn.

In the classical bandit problem answering the exploration-exploitation trade-off essentially consists in sampling some sub-optimal actions from time to time. In an MDP it is no longer so simple: exploration has to be done on actions, but also on states of the environment. This means an agent which wants to explore an action in some state will have to think ahead of how to get into the state in the first place. This is a much more complex problem than in the bandit case, and will feature largely in this paper, notably in chapter 2. On the other hand, learning in an MDP isn't much more complicated as in the bandit case, we just need to learn the parameters of the Markov chain. Overall, the dynamic assumption on the environment changes the how of the exploration-exploitation solution but not the solution itself, unless we are in an adversarial environment where the explore-exploit dilemma doesn't make sense anymore since there is nothing to learn.

There is another distinction about the nature of the environment that is less motivated by intuition, but unfortunately proves to be a major mathematical stumbling block for RL: the nature of the spaces on which we define the MDP. Formally, a MDP is a tuple $(\mathcal{S}, \mathcal{A}, r, P)$ with a state space $\mathcal{S}$ and an action space $\mathcal{A}$ and depends on an external

time-space $\mathcal{T}$. Note that r and P are the reward function and the transition probability kernel. This definition seems innocuous but it leaves the door open for many forms of state-action-time spaces and unfortunately only the situation where they are all finite has been well studied. This is the definition of a *discrete* MDP and is state of the art, but in many problems $\mathcal{S}, \mathcal{A}, \mathcal{T}$ might be countably infinite or even continuous.

It is important to stress that this introduces a difficulty which is poorly understood from a mathematical standpoint. It is not evident that this is an inherent difficulty in the problem of RL unlike, for example, the exploration-exploitation dilemma. The problem rests on questions about how to learn and how to plan in such environment. It is certainly possible to learn how the environment functions even if it is continuous: the tools probably exist already but they need to be adapted, perhaps refined, and applied. Learning a continuous dynamic rather than a Markov chain is more tricky but under some hypothesis it is likely that spleens, Kernel Density Estimation, or similar tools would do the job. Similarly for planning we know that the tools exist in control theory but we need to apply them, and there is here quite some refining to be done. Unfortunately the explore-exploit dilemma is completely uncharted territory in the continuous case, the most there is are some potential hints.

The motivation behind this present line of work is to make progress on continuous time and space RL methods. We think it is necessary particularly for the auction problem faced by Criteo because the space is continuous (any positive price can be set as reserve or bid) and the time scale is effectively countably infinite. We investigate throughout this report, especially in chapter 2, the finite MDP methods and to try and collect together all the tools that exist in RL theory which we will then combine with control theory in future projects to hopefully construct some true continuous time RL methods. Let's not get ahead of ourselves, however: these considerations will be discussed more in depth at the end of this report, for now let's focus on the particular case of RL in repeated auctions.

1.3.2 Application to repeated auction

The previous section outlined the three different axes which differentiate the lines of work in RL: information, type of learning, and nature of the dynamics. However, we can not give an exhaustive literature review of the different combinations in this document. We gave some examples in the previous section, but now let's explore how the buyer and seller problems fall on each of these scales. We will start with the seller problem and then the bidder problem.

In regards to the seller, it is pretty definite that we have a near full information problem. Because the auctions are sealed bids, the seller observes what the bidders have bid, so he observes all the reward he could potentially have recuperated by setting another reserve price. The information is not complete only because it is possible for the seller's reserve to influence the bid distribution whose samples he observes, so there is a sort of

continuous hidden state lurking in the background. However, the effect of this is likely to be very small if the seller is constantly learning, which forces buyers to be near-truthful over time since they can not afford to commit to any very different strategy for extend periods of time.

Both types of learning (online and offline) have been proposed in the literature for learning to set reserve prices. Since offline learning can be viewed as an explore-then-commit strategy in the online learning framework it makes sense to consider it from the online approach. It will offer some nice advantages too: simple efficient updates, adaptability, tracking of non-stationary behaviour which we will discuss more in chapter 3. Overall, we believe it is more naturally understood as an online learning problem even if this is the less common formulation in the literature.

The nature of the dynamics for the seller has been discussed before: bidders may move their state (the strategy) but over long time scales they must remain pretty close to truthful and pretty close to stationary. It does not fall into the nice categories of RL, but since the information is complete it comes down to some more general form of optimisation with feedback. Considering this problem as online rather than offline opens up the elegant domain of online optimisation, which will satisfy all our desires: online stationary Markovian updates to the reserve price which can track non-stationary behaviour and can also converge to optima of stationary distributions.

In this report we will consider two models for the seller: a true, continuous, online optimisation model in chapter 3 where we try and solve the problem from the point of view of the seller and later an approximation which will be the discret MDP environment in which we can test the state of the art RL methods, which only work in the discrete case in chapter 2. For the buyer we will have to commit ourselves to the exact same exercise: one continuous model which we are really trying to solve and a discret approximation to work with and draw inspiration from in order to solve the harder continuous problem.

Since we successfully modelled the seller's dynamic pricing as a stationary Markovian update process, with a more detailed analysis to come in chapter 3, we can formulate the environment for the buyer as an MDP, continuous or its discret approximation. As we have stated repeatedly, we will focus on the discrete approximation since we want to study the current methods to extract their tools and be able to adapt them in the continuous case. So, for practical purposes we have a discrete MDP $\mathcal{M}$ in which our agent repeatedly bids and receives as reward his payoff from the auction. The states of the MDP correspond to the value (draw at random from a value distribution) and the reserve price, and the transitions correspond to the reaction of the seller each time.

The information structure for the buyer is naturally a bandit feedback, since they only observe the reward and transition that actually occur along their trajectory. There is a double restriction on information: first in a given state the buyer only observes the outcome of the chosen action, and second they can not observe anything which depends on the state they are not currently in. In consequence we will face the harder explore-exploit

dilemma which occurs when we have a multi-state MDP environment.

In conclusion, we ponder two practical discret models: a full information online optimisation one for the seller and a discrete multi-state MDP with bandit feedback for the bidder. We know for each one which problems will arise from RL theory, and we will pay special attention to the three main sub-problems: learning, planning, and the balancing of both. We gave the high-level grounding for both these problems and the specific modelling choices we made (ignoring competition, etc.). The rest of this report is divided into three predictable parts: one for the bidder, one for the seller, and in conclusion one which bring the two together to evaluate the methods to provide some insights into the continuous time problem.

Chapter 2

SOTA in discrete RL: UCRL2

In this chapter we will examine the state of the art methods in discrete MDPs which the RL literature has produced. This will help us identify where the methods break down in the continuous case, and also help us identify the tools which RL uses to solve the three component sub-problems. We begin with a deeper examination of the sub-problems of online RL, after which we detail the UCRL2 algorithm and give a novel proof. The interest of this section is mostly about reconsidering existing work under the new introspective lens of trying to solve continuous problems.

2.1 Component sub-problems of online RL

We consider an MDP $\mathcal{M} = (\mathcal{A}, \mathcal{S}, P, r)$ with finite actions and states, and countable time. Our objective is to learn an optimal (vis a vis the objective on r in eq. 1.1) policy π^* for this MDP as effectively as possible. As we saw in section 1.3.1, there are three component problems for online partial information RL problems: learning, planning and balancing exploration-exploitation. In our situation the later two are notably more difficult because of the multiple states of the MDP.

2.1.1 The learning problem

The learning problem as it stands in discrete parametric RL is essentially only an estimation problem. If we are considering a discrete MDP then we know it can be learnt by simply learning the parameters of the Markov Chain, without any need for complex modelling issues. If we did not have this restriction we would need to be more careful and use much more powerful tools. For example, if the dynamic of the environment was controlled by an *a priori* arbitrary differential equation of high dimensional function we would need to employ approximators with many degrees of freedom such as a neural networks.

This is already hinting at the complexity of continuous time/space learning and we

do not want to get ahead of ourselves. The reason for mentioning it already regardless is that we want to make sure the reader takes note that learning is, in general, a non-trivial problem and that it just so happens that parameter estimations from a set of finite samples is so ubiquitous and thoroughly understood in statistics that it seems very trivial in this particular case of the finite MDP.

In essence, all the learning problem involves in a discrete MDP is, again, learning the parameters of the chain which we can do with simple mean estimates. To estimate the reward function at each state action pair simply take the average of rewards collected in that pair in the past. To estimate the transitions from a state-action pair simply take the empirical distribution function over the next states. All of these estimates have finite variance, behave like univariate estimation vis-a-vis confidence intervals and such.

The learning problem is therefore more of a theoretical problem than a real one in the case of discrete MDPs, but this is an artefact of the discrete case. One unfortunate impact of this of course is that we will have a memory footprint for any estimations of the MDP which is linear in the number of actions and quadratic in the number of states. This problem, however docile in the discrete case will return with a vengeance in the continuous case so it can not be forgotten or ignored lightly.

2.1.2 The planning problem

Given an MDP $\mathcal{M}$ is known, which means we know the transitions P and average rewards $\bar{r} = \mathbb{E}[r]$ as well as the sets $\mathcal{A}$ and $\mathcal{S}$, it seems intuitive that we could compute the optimal interaction policy (a.k.a. *control*). This is exactly the planning problem. If we can lay out a simulation of the system at all times, then it stands to reason that we can compute an optimal policy which should be stationary if the environment is stationary.

There are several methods to do this, but we will focus in particular on Value Iteration (VI). This is an approximation scheme so no exact solution will be reached, but it is generally faster than its main alternative (policy iteration) which is exact. In the process of explaining value iteration we will give a natural motivation for using approximation. The core idea of VI is very simple: if one knows the system, one can simulate trajectories from all states and compute their expected reward under a greedy policy. Letting the lengths of the trajectory go to infinity, this approximation should converge correctly to the asymptotical average reward of the optimal policy, which is our desired objective. It will thus have implicitly computed an approximation of the optimal policy, which we can recover.

To perform this "simulation" mathematically we recursively apply a Bellman Equation. Understanding Bellman equations is the key to value iteration and RL, because it is key to control theory, so we will extensively introduce them now, and naturally flow into value iteration. Bellman equations are a foundational concept in reinforcement learning

and this report and Jaksch et al.[17] cannot be understood without them. To begin, equation 2.1 defines the Bellman equation for a policy π, where ρ^π is the asymptotic average reward (our objective) collected by π, and $u^\pi(s)$ is the bias under π of state s. The bias in this case represents the advantage (in term of rewards) one has when starting in s rather than starting in the stationary regime (wrt π) of the *decision* process.

$$\rho^\pi + u^\pi(s) = \bar{r}(s, \pi(s)) + \mathbb{E}_{P^\pi}[u^\pi(s')] \tag{2.1}$$

Equation 2.1 expresses the reward one expects to collect when following policy π for one time step from state s. It is complicated, and in general, when P is a kernel and not a rank 3 tensor as in this discrete case, this become an integral equation in $u(s)$ and is outrageously complicated in terms of π. This is unfortunate since we are actually searching for a policy π to maximise our reward. There is here a clever trick here which involves using the biases u to compute π^*. Note that an optimal strategy[1] π^* should maximise reward, so it will clearly maximise the right hand side. Since it is stationary it will also be equivalent to picking an optimal action for each state, which gives us for each s:

$$\begin{aligned}
\rho^* + u^*(s) &= \bar{r}(s, \pi^*(s)) + \mathbb{E}_{P^{\pi^*}}[u^*(s')] && (2.2)\\
&= \max_\pi \{\bar{r}(s, \pi(s)) + \mathbb{E}_{P^\pi}[u^\pi(s')]\} && (2.3)\\
&= \max_{a \in \mathcal{A}} \{\bar{r}(s, a) + \mathbb{E}_{P^a}[u(s')]\}\,. && (2.4)
\end{aligned}$$

So, in the planning problem, for a known MDP we have characterised an optimal policy with equation 2.4, we have even made a step towards solving it since we now have a maximum over $\mathcal{A}$. The question remains "how to compute π^* given $\mathcal{M}$ and equation 2.4?". The answer for this is to observe that we can rearrange equation 2.4 to form the fixed point of an operator on the biases and thus implicitly on the space of stationary policies by equation 2.3. This fixed point satisfies for all s:

$$u^*(s) = \max_{a \in \mathcal{A}} \{\bar{r}(s, \pi^*(s)) + \mathbb{E}_{P^{\pi^*}}[u^{\pi^*}(s')] - \rho^{\pi^*}\}\,. \tag{2.5}$$

Now, since repeated application of this particular operator converges[2], with some subtleties, we can pick an arbitrary $u_0(s)$ for all s then define the recursion on u:

$$\rho^* + u_{i+1}(s) = \max_{a \in \mathcal{A}} \{\bar{r}(s, a) + \mathbb{E}_{P^a}[u_i(s')]\}\,. \tag{2.6}$$

See that T steps of this value iteration is exactly the same as simulating T steps of a greedy policy in the expectation of the system. This iteration is to be repeated until it converges to the true biases u, which must happen in the limit as $T \to \infty$ since the simulated MDP

[1] Which exists under the condition that the MDP is communicating, which means it has no transient states. An optimal transient state would mean that an optimal policy does not exist since that state can not be reached infinitely many times.

[2] The proof is very cumbersome so we did not reproduce it but readers can find it in Bertsekas[7].

will reach stationarity, where the optimal action collects ρ^* on average regardless of the state so the increments of $u_{i+1} - u_i$ must converge to 0. Now, to obtain an optimal policy from all this, it suffices to take $\pi^*(s) = a^*$ for each $s \in \mathcal{S}$, where a^* is the action that maximises the right hand side of equation 2.6 after it has converged.

The subtleties with value iteration lie in the fact that this recursion on u can not be implemented in practice since ρ^* is unknown (the whole point of this exercise is to compute a policy which collects ρ^* on average). There are two ways around this: the first is heuristic, if we take the stopping condition on the difference of iterates, the terms will cancel, which can also be achieved by relative value iteration; the second is to look at the convergence of the ratio of the u_i/i which converges. Both are covered in more detail in Bertsekas[7], we will settle for the first one, but this doesn't really matter either way for our purpose.

What matters is that we have an iterative procedure to compute optimal policies. This iterative procedure employs the properties of dynamic programming, since it uses the Bellman equation, to control a known system $\mathcal{M}$. It is computationally wasteful as it requires simulations of the whole system, but this is relatively simple in discrete systems since everything can be written as linear algebra, including the expectation, so it only involves a maximum over a vector. However, $\mathcal{M}$ is rarely known in practical reinforcement learning problems outside of very simple games, and it is unknown in the RL formulation we gave in section 1.3.2. We do not know how to control unknown systems, neither in RL nor in control theory, so when evolving in our problem we are going to have to pick an uncertain estimate of the MDP and plan inside it to get a policy to use at the next step. But to get good estimates we must explore, which involves not following the policy. How to adress this problem is the hardest part of RL problems and is the topic of the next section.

2.1.3 Balancing exploration and exploitation

When the MDP is unknown, we do not know how to plan in any meaningful way, so the first question to address in building an algorithm is to decide what MDP should the planning be done in. If we take the current estimate of the MDP we will only be visiting state-action pairs that we already know (since we have a non-zero estimate for them). It is likely that we will only visit other states through random chance of the transitions, and this is not an effective method. If we stick to the empirical estimates, the "learning rate" of the algorithm *i.e.* the rate at which it explores states, will be too low to balance out, in expectation, the generalisation error from planning in the estimate MDP. On the other hand, a pure exploration policy is extremely wasteful since all the random actions will drown out the reward collected by the planned policy, even if it results in a very good estimate of the optimal policy.

These problems are efficiently studied in bandit theory through tools like regrets, and

these notions hold the key to the exploration-exploitation dilemma in the multi-state case too. This is an old and vast topic of research, but we will focus on the classical formulation and its frequentist solutions. In a bandit, one has a single state, but a fixed number of actions (called arms) whose rewards are random. The object, of course, is to learn the best arm. In a bandit, learning $\mathcal{M}$ is just estimating $r(a)$ so the learning problem they pose is very simple it can be solved with MLE (averages) or MAP. Their planning problem is trivial since they are single state, the Bellman equation simply becomes $a^* = \max_a \bar{r}(a)$. But since they are online partial information problems, their study is almost entirely focused on exploration-exploitation.

The particularity of bandits is that one has to learn the system from the inside. In VI we could compute the optimal policy and then run it from round 1 to infinity, in a bandit however at round 1 you can not do better than selecting a random action since the agent has no information about the arms. So there is a cost to pay at each round for not being optimal, but one has to keep taking sub-optimal actions to explore the distributions of rewards and be sure that the current estimates are not statistical flukes. The exploration-exploitation dilemma manifests directly though the main measure used in bandit theory, the regret. It directly measures the efficiency of an online learning procedure in terms of the speed at which it's collected reward converges to the optimal strategy in hindsight.

The observation made by bandit theorists very early on is that we only care about maintaining the accuracy of actions which are likely to actually be good. Very sub-optimal actions do not help improve the regret so they are of no use. It isn't only about uncertainty but also about the uncertainty of the estimates relative to each other. The way to incorporate both these observations into a bandit algorithm is to look at the confidence intervals of each action, and compare their upper bounds. If one action has a high mean with high confidence, but the upper confidence bound on another action is higher, we should sample the second action to check wether it actually is the best. This algorithm is called the Upper Confidence Bound (UCB) algorithm, and it is the direct inspiration of UCRL or Upper Confidence for Reinforcement Learning. In essence, bandit theory states that the way to choose when to explore and when to exploit is a paradigm of *optimism*: act as if the best outcome is always going to occur.

To temper this optimism there are some mathematical tools, which again are readily available in UCB. First, to build the confidence intervals, we can use Chernoff-Hoeffding confidence bounds, and a global parameter δ which ensures that the confidence intervals of each arm are directly comparable. Then, we want to use in the bound of each action the number of times it has been sampled to incite exploration of arms which are under represented. Otherwise, we would be neglecting to compare the uncertainties between arms. In the end, carefully tempered optimism can allow us to minimise our regret, *i.e.* the loss relative to an optimal policy. Let's introduce these tools to conclude this overview of the exploration-exploitation dilemma.

We presented the notions of optimism above under the guise of the bandit case, but this is slightly complicated by the multi-state problem of UCRL2. As an example of

this, consider the regret comparator: should we compare ourselves the the optimal policy along the same path of states that was generated by our algorithm, or should we compare ourselves to the optimal policy with the transition probabilities it induces along its own (optimal) path. A bit of thought will show that the later is what we want, since it will give a definition of the regret as

$$R(T) := \sum_{t=1}^{T} \rho^* - r_t \,. \tag{2.7}$$

And it is indeed ρ^* that we seek to attain by maximising our objective of average reward (eq. 1.1). This particular example was easily solved, but it offers a glimpse to the non-trivial changes that modelling several states induces.

In order to define the confidence sets we will follow exactly the same steps as in the stochastic bandit case, but now with all the different parameters of the MDP. Consider a parameter θ bounded in $[0, 1]$ whose empirical mean with n iid samples is $\hat{\theta}_n$, we can bound the uncertainty on θ without any a priori on its distribution, except that it has finite variance and mean, using Hoeffding's inequality:

$$\mathbb{P}\left(\left|\hat{\theta}_n - \theta\right| \geq \epsilon\right) \leq 2\exp\left(-\frac{2\epsilon^2}{n}\right) .$$

Fixing a confidence level δ, we can construct a confidence interval since with probability at least δ

$$\left|\hat{\theta}_n - \theta\right| \leq \sqrt{\frac{1}{2n}\log(2/\delta)} \,.$$

We construct such an interval for the mean rewards $\bar{r}$, estimated by $\hat{r}$, for each state action pair. We then denote $\mathcal{C}^r$ the confidence set defined by all vectors r whose components satisfy all the individual confidence intervals. This constitutes a confidence set for rewards. The same process applies for transition probabilities but with triples of two states and an action instead of state action pairs, and we denote the resulting set $\mathcal{C}^P$. Let $\mathcal{C}$ denote the set of discrete communicating MDPs whose parameters satisfy both $\mathcal{C}^P$ and $\mathcal{C}^r$, it corresponds to the confidence set in the space of MDPs defined by a range δ around the point-estimates. This corresponds to UCB's confidence intervals, and by analogy what remains to make our optimistic decision and solve the exploration-exploitation dilemma is to choose $\tilde{\mathcal{M}} \in \mathcal{C}$ so that $\tilde{\mathcal{M}}$ maximises $\rho^*(\tilde{\mathcal{M}})$.

This maximisation exercise is highly non-trivial and it is *a priori* intractable. The beautiful trick of UCRL2 is to offer a simple procedure to do this maximisation, which means a simple way to optimistically explore the space, which will solve the exploration-exploitation dilemma like in a bandit. Together with classical point estimates from statistics to address the learning problem and value iteration from control theory to address the planning, we have presented all the tools needed to put together the UCRL2 algorithm. In the next section we will look to present the algorithm and give it a rigourous mathematical analysis.

2.2 The UCRL2 algorithm

2.2.1 Outline

We give the detailed steps of the algorithm in algorithm 2.2.1, which the reader can consult if they want all the details. The text of this section takes a slightly higher level of analysis by breaking the algorithm into its component sub-routines rather than component steps. In the next section we will analyse the little details and properties that all the algorithm to function smoothly, which will be much easier to comprehend after some high level explanations about the sub-routines and the main contribution of the algorithm which is Extended Value Iteration (EVI). Time is divided into episodes in UCRL2, and we now list the sub-routines in order of execution from the start of an episode.

1. At the beginning of each episode, the algorithm runs an initialisation routine. Its purpose is just to update stored variables which are needed later in the algorithm. First, it computes and stores accumulated rewards, visit counts and transition counts, then estimates rewards $\hat{r}$ and transitions $\hat{P}$ for each state-action pair or triple. This step is purely programatic, and is just the learning/estimation implementation.

2. Now that it has empirical estimates and observed quantities on hand, the algorithm enters a second subroutine which computes the set of all acceptable MDPs. Explicitly it computes the bounds on $\tilde{r}$ and $\tilde{P}$ that define the confidence set $\mathcal{C}$ along with the natural constraints (*e.g.* $\tilde{r} \in [0,1]^{\mathcal{S}\times\mathcal{A}}$). This is the confidence set over which we want to perform our optimistic choice of $\tilde{\mathcal{M}}$. The details of this routine are important for the computation of the regret, but on a high level it is just the aggregation of a bunch of linear Hoeffding inequalities. We will denote the set of MDPS that satisfy the constraint $\mathcal{C}$, which we decompose into $\mathcal{C}^P$ the set of plausible transitions and $\mathcal{C}^r$ the set of plausible mean rewards.

3. Now, we find the optimistic MDP and plan its optimal strategy. This is the core of the algorithm and is called Extended Value Iteration (EVI). We will describe its details extensively shortly, for now let's give a first high level explanation. Remember that classic value iteration allows us to find the value of each state and plan from this an optimal policy by maximising over the set of actions of the MDP and that optimism is maximising expected reward over candidates. The trick here is to create a meta-MDP $\mathfrak{M}$ whose action set is $\mathcal{A}\times\mathcal{C}$, so that choosing a policy in $\mathfrak{M}$ is equivalent to choosing an MDP in $\mathcal{C}$ and a policy inside it to perform both maximisations at once. It may seem surprising, but performing Value Iteration in $\mathfrak{M}$ corresponds to optimistically choosing an MDP in $\mathcal{C}$ and then doing value iteration inside it. A comfortable reader might agree already, but we will give this argument rigorously in section 2.2.2.

4. The first three routines are run once at the start of the episode, and together compute a policy. The final routine executes this policy for the remainder of the duration of the episode and decides when to terminate the episode. Obviously, one needs to run the policy to collect new data and improve it, but the choice of termination condition is not obvious. It was chosen in this algorithm that if a state-action is visited as often this epoch as in all previous ones combined, the epoch will terminate. We will clarify later this particular choice, but the reader should note that it is far from innocuous. This termination condition is the detail to remember about this subroutine, as the rest is just execution of a policy.

In this outline we singled out the key component of UCRL2, extended value iteration, which allows us to do apply optimistic learning to unknown MDPs and optimal planning at the same time. We will detail it right away in the next subsection. In regards to the epoch termination condition and the particular bounds chosen for $\mathcal{C}$ these will be explained in section 2.3 as they warrant special attention to guarantee a good regret but they are not the essence of the algorithm like EVI.

2.2.2 The Key trick: Extended Value Iteration

When we introduced extended value iteration above in item 3, we claimed that value iteration in a meta-MDP was the same as optimistic choice of an MDP and value iteration inside it. This is a conceptually challenging assertion but its proof is quite simple if the original MDP is discrete (which is the case here). First, we should convince the reader that for any meta-MDP $\mathfrak{M}$ and policy π_m there is an MDP $\tilde{\mathcal{M}} \in \mathcal{C}$ and a policy $\tilde{\pi}$ which induces the same transition probabilities and average rewards. Note that the action set of $\mathfrak{M}$ is $\mathfrak{A} := \mathcal{A} \times \mathcal{C}^P \times \mathcal{C}^r$ while its state space is $\mathcal{S}$, so we can decompose its policy into three components and look at the effect of each one. See that the part of π_m which applies to true actions $\mathcal{A}$ (denoted π_{ma}) can be explicitly reproduced in $\tilde{\mathcal{M}}$. So really, it is just a matter of seeing that π_{mP} and π_{mr} can be reproduced by $\tilde{\mathcal{M}}$. Clearly, we can take $P^\pi = \pi_{m_P^\pi}$ as the TPM induced by π, and we can set any remaining values of the transition probability tensor of $\tilde{\mathcal{M}}$ with their estimates $\hat{P}$. By definition of $\mathfrak{M}$ and as $\hat{p} \in \mathcal{C}$ trivially this MDP $\tilde{\mathcal{M}}$ is in the candidate set. The reader can check that we can proceed likewise for r. This isn't much of an issue, but it is important to note that this trick is justified.

Now that we've established that this move from $\mathcal{M}$ to $\mathfrak{M}$ even makes sense, we should explain how EVI actually works. As we stated in the previous subsection, it is value iteration in $\mathfrak{M}$. Let's write out the optimal Bellman equation to see what is happening:

$$\rho^* + u_{i+1} = \max_{(a,\tilde{P},\tilde{r})\in\mathfrak{A}} \{\mathbb{E}_{\tilde{P}}[\tilde{r}(s,a) + u_i]\}\,.$$

This maximisation problem doesn't appear tractable as such, but we can separate the

components of a, P, and r:

$$\rho^* + u_{i+1} = \max_{a\in\mathcal{A}}\{\max_{\tilde{r}\in\mathcal{C}^r}\{\tilde{r}(a)\} + \max_{\tilde{P}(\cdot|a)\in\mathcal{C}^P}\{\mathbb{E}_{\tilde{P}}[u_i]\}\}\,.$$

This seems more promising, we can perform the maximisation or $\tilde{r}$ trivially, and since the number of states and actions is finite we can decompose the last term from an expectation to a linear program:

$$\rho^* + u_{i+1} = \max_{a\in\mathcal{A}}\{\tilde{r}(a) + \max_{\tilde{P}\in\mathcal{C}^P}\{\tilde{P}(\cdot|a)\cdot u_i\}\}\,.$$

The last thing to notice is that $\mathcal{C}^P$ (resp. $\mathcal{C}^r$) is a convex polytope as it is defined by linear constraints intersecting a simplex (resp. a hypercube). Minimising a linear program over this convex polytope only requires us to check a finite number of vertices $\mathfrak{V}$ and we can leverage efficient existing tools. For example, we can rewrite this as value iteration in a finite communicating MDP:

$$\rho^* + u_{i+1} = \max_{a\in\mathcal{A}\times\mathfrak{V}}\{\tilde{r}(a) + \mathbb{E}_{P(\cdot|a)}[u_i]\}\,.$$

It would also be possible and sometimes more feasible to use LP solvers directly; the key take-away is that we have reduced a very hard optimisation problem to a relatively simple one.

To reiterate, we've shown that EVI does induce an MDP in the candidate set, and also that it boils down to VI over a particular MDP with finite states or to an LP, so that it is solvable and guaranteed to converge. This convergence is asymptotic however and we must choose an approximation criterion to interrupt the process. Since at time $t = t_k$ the random noise in the estimates is $\mathcal{O}(1/\sqrt{t_k})$ we can take

$$\max_{s\in\mathcal{S}}\{v_i(s) - v_{i-1}(s)\} - \min_{s\in\mathcal{S}}\{v_i(s) - v_{i-1}(s)\} < \frac{1}{\sqrt{t_k}}$$

as the termination condition. Note that this rids us of ρ^* as we outlined in standard Value Iteration. This condition simply requires uniform updating of the values by an increment decreasing as $1/\sqrt{t}$.

Extended Value Iteration is really the heart of the UCRL2 algorithm, and it is what makes it implementable whilst the original UCRL[4] only worked on paper. Boiling down optimistic decision making in an unknown environment to classic value iteration and a linear program is very significant. What is even more surprising is that the components of Bandit theory and RL which we considered opposites in the introductory section are very closely linked and generalise jointly in a completely natural way. Optimistic choice in unknown systems is a value iteration. This synthesis shows that the problem is harder than classical RL, but not as much as one could have feared. If the reader has understood EVI they have essentially understood the whole algorithm on at least the surface level. In the next section we propose to enter the nitty gritty details of the algorithm's particular features. This is interesting in order to develop an intimate understanding of the program but it will also allow us to present mathematical tools for the analysis of the regret of UCRL.

2.3 Properties of the UCRL2 algorithm

2.3.1 General properties and lemmas

While EVI was given its own presentation in the previous section the purpose of this section is to extract further meaning from the overall algorithm and its details. All of these facts will be used in the proof of a regret bound, which is why they are presented as lemma and not remarks or properties, but they are very much intrinsic properties of the algorithm. We will begin by the highest level explicit statements and move slowly towards more abstract results which constitute mathematical tools. First, we'll look at the number of epochs, then we'll look at some properties of candidate MDPs (and of the true one too). Then we propose some combinatorial lemmas and probabilistic results which will help us bound regret terms later. Finally, we will examine the impact on the regret of the probability that we miss out on $\mathcal{M}$ in one of our confidence sets.

Lemma 1 tells us that the condition for epoch termination induces an epoch size essentially exponential in T, and thus that there are at most a logarithmic number of them before a given time. In particular, see that the condition requires a doubling of the visit count of a state-action pair once each state-action pair has been visited once. So, we will have a constant term for the "burn-in" and then a $\mathcal{O}(\log_2(T))$ term for the logarithmic phase. This is an important property for UCRL, as if we allow the number of epochs to be too large (if they have constant length it would be $\mathcal{O}(T)$ for example) then the algorithm can become indecisive and dither.

Lemma 1 (Episode count of UCRL2)**.** *For all $T > SA$, the number of epochs up to T of UCRL $m(T)$ satisfies:*

$$m(T) \leq 1 + 2SA + SA \log_2 \frac{T}{SA}\,.$$

Proof of lemma 1. The proof is not particularly interesting, it just puts into algebra the distinction made above, and then some minor manipulation yields the results.

There are two ways an epoch can terminate: either there is a state-action pair (s, a) whose count doubled (*i.e.* $v_k(s, a) = N_k(s, a)$), or we had $N_k(s, a) = 0$ and we just visited (s, a) for the first time. In the former case, let us define $K(s, a)$ the number of episodes terminated by triggering the condition $v_k(s, a) = N_k(s, a)$. For $N(s, a) > 0$, which is a given in this case, we have:

$$N(s,a) = \sum_k v_k(s,a) \geq 1 + \sum_{k : v_k(s,a) = N_k(s,a)} N_k(s,a)\,.$$

Since the terms in the rightmost sum are exactly the counts which double, we can change the indexation and write it as:

$$1 + \sum_{i=1}^{K(s,a)} 2^{i-1} = 2^{K(s,a)} .$$

In the second case, we have $N(s,a) = 0$ which implies $K(s,a) = 0$, so we can generalise our inequality to this case with:

$$N(s,a) \geq 2^{K(s,a)} - 1 .$$

Summing over (s,a), we have $T = \sum_{(s,a)} N(s,a) \geq \sum_{(s,a)} 2^{K(s,a)} - 1$. Now, let us translate this in terms of m as for each episode k there is exactly one (s,a) which causes its termination, and there are at most SA epochs which terminate as $v_k(s,a) = 1$, thus we have:

$$m \leq 1 + SA + \sum_{(s,a)} K(s,a) .$$

Let's combine these two results:

$$\begin{aligned} T &\geq \sum_{(s,a)} 2^{K(s,a)} - 1 \\ &\geq SA \left(2^{\sum_{(s,a)} \frac{K(s,a)}{SA}} - 1 \right) \\ &\geq SA \left(2^{\frac{m(T)-1}{SA} - 1} - 1 \right) \end{aligned}$$

Solving for m, we obtain the result:

$$m(T) \leq 1 + 2SA + SA \log_2 \frac{T}{SA} .$$

□

What the reader should take away from this lemma is that the number of epochs grows essentially logarithmically with time as long as the number of time steps is notably larger that SA, which is the domain where the problem is meaningful. The next couple of lemmas are about the variation in value both within and between MDPs susceptible of being chosen. First we'll see that a communicating MDP has a natural bound of the difference in the bias u (the finite time advantage of starting in a given state) between states. Next, we'll prove that if the true MDP is a candidate, then the optimistically chosen MDP must be communicating and have a smaller diameter. Together, this will allow us to identify the meaning of optimism on the transition probabilities.

Lemma 2. *Define $w(s) := u(s) - \min_s u(s)$, then we have in a communicating MDP with diameter D with rewards in $[0,1]$, that:*

$$\|w(s)\|_\infty \leq D .$$

This is implicitly a bound on the *span* of the MDP, defined as $\max_s u(s) - \min_s u(s)$, stating that the value of any state can't be suboptimal by more than D. Intuitively this lemma is a direct consequence of the communication property. This is quite simply because if one can move from any suboptimal state to the optimal one (in expectation) in D steps then one loses out on at most D reward during the movement. So the difference in value by the Bellman equation is less than D. Below is a formal proof, which follows this intuition:

Proof of lemma 2. Let's take a Markov Decision Process, where the state of highest value is s^+ and the state of lowest value is s^-. Let us consider two policies, π^* the optimal policy and π' a policy which moves as fast as possible in expectation from s^- to s^+ then follows π^* forever. Note that formally:

$$\pi' \in \operatorname*{arginf}_{\pi} \mathbb{E}[\inf_t \{t \,:\, s_t = s^+\}] = \operatorname*{argmin}_{\pi} \mathbb{E}[\inf_t \{t \,:\, s_t = s^+\}]\,,$$

as this infimum is attained. Let's now focus on the difference in value between $u^+ = u(s^+)$ and $u^- = u(s^-)$, *i.e.* the span of the MDP. Let's take the Bellman equation for π^* and π' in s^-, we have:

$$\begin{aligned}\rho^* + u^*(s^-) &\geq \rho' + u'(s^-) \\ &\geq r(s^-, \pi'(s^-)) + \mathbb{E}[u'(s')|s' \sim P']\,.\end{aligned}$$

Let's now apply the Bellman operator recursively, until a stopping time τ is reached. We define τ as the first time we reach s^+ when starting from s^- following π'. This yields:

$$\rho^* + u^*(s^-) \geq \mathbb{E}\left[\sum_{t=1}^{\tau} r(s_t, \pi'(s_t))|s_0 = s^-,\, s_t \sim P'(s_{t-1})\right] + \mathbb{E}[u^*(s_{\tau+1})]\,.$$

Now, let's compare this with what we would get if we started in s^+ and just stayed with π^*:

$$\begin{aligned}u^*(s^+) - u^*(s^-) &\leq \mathbb{E}\left[\sum_{t=1}^{\tau} r(s_t, \pi^*(s_t))|s_0 = s^+,\, s_t \sim P^*(s_{t-1})\right] + \mathbb{E}[u^*(s_{\tau+1})] \\ &\quad - \mathbb{E}\left[\sum_{t=1}^{\tau} r(s_t, \pi'(s_t))|s_0 = s^-,\, s_t \sim P'(s_{t-1})\right] - \mathbb{E}[u^*(s_{\tau+1})] \\ &\leq \mathbb{E}\left[\sum_{t=1}^{\tau} 1\right] - \mathbb{E}\left[\sum_{t=1}^{\tau} 0\right] \\ &\leq \mathbb{E}[\tau]\,.\end{aligned}$$

See that by the definition of π', we have:

$$\mathbb{E}[\tau] = \inf_{\pi} \mathbb{E}[\inf_t \{t \,:\, s_t = s^+,\, s_0 = s^-\,, (\forall i < t)(s_{i+1} = P^\pi(s_i))\}]\,.$$

We can maximise over state pairs, and this recovers the definition of the diameter, proving the result.

$$u^*(s^+) - u^*(s^-) \leq \max_{s,s'} \min_{\pi} \mathbb{E}[\inf_t \{t \,:\, s_t = s,\, s_0 = s'\,, s_{i+1} = P^\pi(s_i)\}] = D\,.$$

□

In the previous lemma, we showed that the span of a communicating MDP M was upper bounded by its diameter. However, when picking an MDP $\tilde{M}$ with UCRL2, we do not know a priori that it will be communicating or what its diameter or span will be. The next lemma will show that any MDP chosen by UCRL2 will necessarily have span less than the diameter of the true MDP which means it is communicating, and thus that we can bound its span as well. This means that optimism about the transition probabilities is both hoping that as much mass is put on the state of highest value as possible, but also that it is easy to move between states.

Lemma 3. *If the true MDP is in the confidence set, then the meta-MDP $\mathfrak{M}$ is communicating with diameter at most D. As a consequence, the span of $\tilde{\mathcal{M}}$ is less than D.*

Proof of lemma 3. The first statement requires us to show that the diameter of the meta-MDP $\mathfrak{M}$ is less than the diameter of the true MDP, which is D, so long as $\mathcal{M}$ is a valid candidate MDP. We have:

$$\begin{aligned}\mathrm{Diam}(\mathfrak{M}) &= \max_{s,s'} \min_{\pi_m=(\pi,r,P)} \mathbb{E}_{P^\pi}[\inf\{t \,:\, s_t = s',\, s_0 = s\}] \\ &\leq \max_{s,s'} \min_{\pi} \min_{r\in\mathcal{C}^r} \min_{P\in\mathcal{C}^P} \mathbb{E}_{P^\pi}[\inf\{t \,:\, s_t = s',\, s_0 = s\}]\,.\end{aligned}$$

The minimisation on r has no effect on the time to move between states, so we can ignore it. On the other hand, since $\mathcal{M} \in \mathcal{C}$ by the conditions of the lemma, we can upper bound the inner minimum by substituting the true probability tensor P. This will recover the definition of the diameter of $\mathcal{M}$ and complete the proof:

$$\begin{aligned}\mathrm{Diam}(\mathfrak{M}) &\leq \max_{s,s'} \min_{\pi} \mathbb{E}_{P^\pi}[\inf\{t \,:\, s_t = s',\, s_0 = s\}] \\ &\leq \mathrm{Diam}(\mathcal{M}) = D\,.\end{aligned}$$

Now, observe that the biases of $\mathfrak{M}$ as computed by EVI are also the biases of $\tilde{\mathcal{M}}$. Since $\mathrm{Diam}(\mathfrak{M}) \leq D$, applying lemma 2 completes the proof of the second half. □

With these two lemmas, we can affirm that the span of $\tilde{\mathcal{M}}$ generated by UCRL2 is always bounded above by D. We will put this to use several times in the proof by bounding $\|\tilde{w}_t\|_\infty$ by D. The important intuition to take away from these two lemmas is that UCRL2 will choose (so long as it can) MDPs in which it is easy to move to zones of

high reward, which is the condition to have low span. Indeed, the most optimistic it can be consists in assuming a near-treelike structure to the MDP where all states flow with maximal probability towards the optimal one, which is near absorbent.

Explicitely we will use this new result in lemma 4 to construct a Martingale Difference Sequence (MDS) to which we will apply Azuma's inequality later on in the proof of this paper's main theorem.

Lemma 4. *The sequence* $\{Y_t\}_t$ *where* $Y_t := \mathbb{E}_P[w_t(s')|\mathcal{F}_{t-1}] - w_t(s_t)$*, with* $\|w\|_\infty \leq D$*, is a martingale difference sequence.*

Proof of lemma 4. It is somewhat trivial to see that Y_t is $\mathbb{F}$-adapted where $\mathbb{F}$ is the natural filtration. To show that the Y_t are absolutely summable, it suffices to note we can upper bound them as follows:

$$Y_t \leq \mathbb{E}[\|\tilde{w}_t\|_\infty] + \|\tilde{w}_t\|_\infty \leq 2D\,.$$

It also trivial that $\mathbb{E}[Y_t|\mathcal{F}_{t-1}] = 0$ simply from the definition of Y_t, so $\{Y_t\}_t$ is a MDS. □

This lemma bounds the size of deviations in $\tilde{w}$ between the expected trajectory and the travelled trajectory. A combinatorial lemma we will need in the regret is given in lemma 5. It relates the number $v_k(s,a)$ of visits to (s,a) in one epoch to $\max\{1, N_k(s,a)\}$ and thus to the confidence set $\mathcal{C}$.

Lemma 5. *The UCRL2 algorithm, for all* k, s, a*, satisfies:*

$$\sum_k^{m(T)} \sum_{(s,a)} \frac{v_k(s,a)}{\sqrt{\max\{1, N_k(s,a)\}}} \leq (\sqrt{2}+1)\sqrt{SAT}\,.$$

Proof of lemma 5. First, let us state another lemma, and then it will only remain to use it and do some minor algebra to conclude. Its proof is given in the paper for UCRL2 and presents no difficulty: it is simply an induction on n.

Lemma 6 (Self-normalisation lemma)**.** *For any sequence of numbers* $\{x_i\}_{i=1}^n$*, with* $0 \leq x_k \leq X_{k-1} := \max\{1, \sum_{i=1}^{k-1} x_i\}$*:*

$$\sum_{k=1}^{n} \frac{x_k}{\sqrt{X_{k-1}}} \leq (\sqrt{2}+1)\sqrt{X_n}\,.$$

Let's thus start by applying this lemma to our double sum, by taking $X_0 = 1$ so that $X_k := \max\{1, N_k(s,a)\}$ and thus we have: $X_n = \sum_k v_k(s,a) = N(s,a)$ the total number of visits to (s,a). So by applying the lemma and summing over state-action pairs we have:

$$\sum_{(s,a)} \sum_{k}^{m(T)} \frac{v_k(s,a)}{\sqrt{\max\{1, N_k(s,a)\}}} \leq (\sqrt{2}+1) \sum_{(s,a)} \sqrt{N(s,a)}\,.$$

Now, simply applying Jensen's inequality to the right-hand side and upper-bounding the sum by the product of terms we have the desired result:

$$\sum_{(s,a)} \sum_{k}^{m(T)} \frac{v_k(s,a)}{\sqrt{\max\{1, N_k(s,a)\}}} \leq (\sqrt{2}+1)\sqrt{SAT}\,.$$

□

In order to be able to use theses lemmas, we would like to start our proofs by assuming that at each t, $\mathcal{C}_t$ contains the true MDP $\mathcal{M}$, which would allow us to bound terms involving differences in $\tilde{r}$ and r or $\tilde{P}$ and P using the conditions that define $\mathcal{C}$. To do so, we need to show that it holds with high probability on the one hand, and on the other that even if it doesn't hold, the incurred regret is not catastrophic so that the expected regret is not affected dramatically.

Lemma 7. *With probability less than $1 - \frac{\delta}{15T^5}$, the regret incurred in steps where $\mathcal{M}$ was not in $\mathcal{C}$ can be upper bounded by $\sqrt{T}$.*

Proof of lemma 7. Let us define the following events $\mathcal{E}_t = \{\mathcal{M} \in \mathcal{C}_t\}$, and $E_t = \{\bigcap_{s \leq t} \mathcal{E}_t\}$. We will bound the probability that $\mathcal{E}_t$ happens, after which we will build Union bounds until we can bound $\mathbb{P}(E_T^c)$, the probability that $\mathcal{M}$ was not in $\mathcal{C}$ at least once during the whole run, with a small δ. This will lead us to bounding the regret if E_T^c happens, which will thus bound the regret of times where $\mathcal{E}_t^c$ happens. Fix $n = N_k(s,a)$ for each k and (s,a), which we will undo later with a union bound over n. See that:

$$\begin{aligned}
\mathbb{P}(\mathcal{E}_t | \mathcal{F}_{t-1}) &= \mathbb{P}\left(\bigcap_i \{r_i \in \mathcal{C}_t^{r_i}\}\right) \cap \mathbb{P}\left(\bigcap_i \{P_i \in \mathcal{C}_t^{P_i}\}\right) \\
&\geq 1 - \mathbb{P}\left(\bigcap_i \{r_i \notin \mathcal{C}_t^{r_i}\}\right) + \mathbb{P}\left(\bigcap_i \{P_i \notin \mathcal{C}_t^{P_i}\}\right) \\
&\geq 1 - \sum_{i=1}^{SA} \mathbb{P}(\{r_i \notin \mathcal{C}_t^{r_i}\}) - \sum_{i=1}^{SA} \mathbb{P}(\{P_i \notin \mathcal{C}_t^{P_i}\})\,.
\end{aligned}$$

Now what needs to be done is to bound each of these probabilities, which we can do easily by the definitions of the constraints $\mathcal{C}^r$ and $\mathcal{C}^P$ from the algorithm and by using Hoeffding's

inequality. In order to obtain clean terms we will take values of ϵ less than those given in the algorithm. For rewards, let $\epsilon := \sqrt{\log(120SAt^7/\delta)/2n}$. Now see that:

$$\begin{aligned}\mathbb{P}(\{r_i \notin \mathcal{C}_t^{r_i}\}) &= \mathbb{P}\left(|r(s,a) - \hat{r}_k(s,a)| > \sqrt{\frac{\log\left(\frac{120SAt^7}{\delta}\right)}{2n}}\right) \\ &\leq \exp\left(-2n\epsilon^2\right) = \frac{\delta}{60SAt^7}\,.\end{aligned}$$

Now, let's apply a union bound over n, then simply sum over (s,a), which gives a bound of:

$$\mathbb{P}\left(\{r \notin \mathcal{C}_t^r\}\right) \leq \frac{\delta}{60t^6}\,.$$

In very much the same way for P, let us pick $\epsilon' = \sqrt{2\log(2^S 20SAt^7/\delta)/n}$. We will apply another inequality for bounding discrepancies in the one-norm, which can be found in Weissman et al.[43] which gives for $\epsilon > 0$:

$$\mathbb{P}\left(\left\|\hat{P} - P\right\|_1 \geq \epsilon\right) \leq (2^S - 2)\exp\left(\frac{-n\epsilon^2}{2}\right)\,.$$

Discarding the -2 term and simplifying, we recover the following bound:

$$\mathbb{P}\left(\{P_i \notin \mathcal{C}_t^{P_i}\}\right) = \mathbb{P}\left(\left\|\hat{P} - P\right\|_1 \geq \epsilon'\right) \leq \frac{\delta}{20SAt^7}\,.$$

Applying the union bound on n and the sum over (s,a) recovers:

$$\mathbb{P}(\{P \notin \mathcal{C}_t^P\}) \leq \frac{\delta}{20t^6}\,.$$

Summing the probability from the rewards and the transition probabilities, we have $\mathbb{P}(\mathcal{E}_t^c|\mathcal{F}_{t-1}) \leq \frac{\delta}{15t^6}$. Now, to finish we can apply a union bound over $t \leq T$ to get:

$$\mathbb{P}(E_T) \geq 1 - \frac{\delta}{15T^5}\,.$$

We have bounded the probability that $\mathcal{C}$ always contains the true MDP, what remains now is to turn this into a regret bound when it fails. Let's denote by Δ_t the instantaneous regret incurred at time t. We can express the regret of UCRL2 as:

$$R_T = \sum_t [\Delta_t|\mathcal{E}_t]\mathbb{P}(\{\mathcal{E}_t\}) + \sum_t \Delta_t \mathbb{P}(\{\mathcal{E}_t^c\})\,.$$

The first term is upper-bounded by the regret under the assumption that E_T happens, while the second term we can bound using the bond we gave above and a union bound as $\delta/15T^4$ since the Δ_t are in $[0,1]$. Likewise we can express the expected regret of UCRL2 as:

$$\begin{aligned}\mathbb{E}[R_T] &= \sum_t \mathbb{E}[R_T|\mathcal{E}_t]\mathbb{P}(\{\mathcal{E}_t\}) + \sum_t \mathbb{E}[R_T|\mathcal{E}_t^c]\mathbb{P}(\{\mathcal{E}_t^c\}) \\ &\leq \sum_t \mathbb{E}[R_T|\mathcal{E}_t] + \frac{\delta}{15T^4}\,.\end{aligned}$$

□

This lemma justifies our regret analysis of the regret when E_t holds, as with high probability it does and we can guarantee that when it does not, only a few $\mathcal{E}_t$ will fail and they won't induce an instantaneous regret high enough to perturb the expectation of the overall regret. We now have all the tools required to begin an analysis of the high-probability regret, assuming E_T happens, which we can do with a union bound on its probability at the end.

2.3.2 Regret bound & proof sketch

Theorem 8 gives the high-probability worst case regret bound we will prove for UCRL2. It is to be contrasted with theorem 9 and theorem 10 which are the expected (instance dependent) regret and minimax regret respectively. We will discuss them briefly, then give the proof sketch of theorem 8, and finally its detailed proof.

Theorem 8 (High probability worst case regret upper bound for UCRL2)**.** *If UCRL2 is run with confidence parameter δ, for all $s_0 \in \mathcal{S}$ and all $T > 1$ then we have:*

$$R_T \leq CDS\sqrt{AT \log \frac{T}{\delta}}\,.$$

Theorem 9 (Distribution dependent regret upper bound for UCRL2)**.** *If UCRL2 is run with confidence parameter δ, for all $s_0 \in \mathcal{S}$ and all $T > 1$ then we have:*

$$\mathbb{E}[R_T] \leq C' \frac{D^2 S^2 A \log T}{\Delta^*}\,.$$

Where Δ^* is the smallest sub-optimality gap.

Theorem 10 (Minimax Regret bound for learnable MDPS)**.** *For any algorithms there is an MDP under certain size requirements on T, S, A, D (which are satisfied by very large values), such that the following hold:*

$$\mathbb{E}[R_T] \geq C''\sqrt{DSAT}\,.$$

The regret of UCRL2 depends, unsurprisingly on S and A, but also on D. So, the difficulty (as outlined by the minimax bound) is equally contributed to by the size of the MDP, the range of actions one can choose from but also how easy it is to move around the space. The first two are very trivial observations, but the dependence on diameter highlights the issue of exploration and exploitation one faces in unknown MDPS: getting

around the space to explore the states one desires may be quite painful. And the more painful it is, the more difficult it is to learn an optimal strategy.

Note that in all cases the problem is learnable, *i.e.* the regret is sub-linear so that the average reward collected converges to the optimal reward as $T \to \infty$. We do well in expectation, and in the worst case we pay only $\mathcal{O}(\sqrt{DS \log T})$ more than the optimum (the minimax), which is not perfect but not catastrophic either. In the perspective of our research project however, the linear or square root scaling in states and actions is meaningless in continuous MDPs. This is why UCRL2 is not applicable in its form to our problem. There would also be computational problems of course, but even on paper UCRL2 is not fit for very large action and state spaces, let alone continuous space and time.

Nevertheless, let's break down the regret into a decomposition and prove it. This is a different proof from the one of Jaksch et al. and its purpose is to make clear the contributions of different types of errors. Ultimately the hope is that we can use this decomposition to extract meaning from the algorithm and leverage it when we are faces with the problem in the continuous case.

Before anything else, let's fix some notations. At each time t, in epoch k, the transitions probabilities induced by $\pi_t = \pi_k$ are denoted $P_t = P_k = P^{\pi_k}$ for the true MDP M and $\tilde{P}_t$ for the current estimate $\tilde{\mathcal{M}}$. We will denote r_t realisations of the rewards in the true MDP, with mean $\bar{r}$ and $\tilde{r}_t$ the mean rewards of $\tilde{\mathcal{M}}$.

We sketch briefly the proof here, before giving the details. To do so we will decompose the regret into multiple terms, and explain each one now, before proving the bounds on each of them in the next section. The first thing we'll do is separate the term that comes from the variation of the rewards, which by Hoeffding is $\mathcal{O}(\sqrt{T \log T})$. Then, we can work from the expected rewards instead of the random variables, and we can use the Bellman equation to decompose the regret. Let us assume that with probability $1 - \delta'$, at every step the true MDP is in the candidate set (recall lemma 7). Let R_t be the regret, with probability $1 - \delta/(12T^{5/4})$ we would have:

$$
\begin{aligned}
R(s_1, T) &= T\rho^* - \sum_t r_t \\
&\leq \sum_t \rho^* - \bar{r}(s,a) + \sqrt{\frac{5}{2} T \log \frac{8T}{\delta}} \\
&\leq \sum_t \rho^* - \tilde{\rho}_t && (2.8) \\
&\quad + \sum_t \mathbb{E}_{P_{k(t)}}[\tilde{u}_{t+1}(s_{t+1})] - \tilde{u}_t(s_t) && (2.9) \\
&\quad + \sum_t \mathbb{E}_{P_{k(t)}}[\tilde{u}_t(s_{t+1}) - \tilde{u}_{t+1}(s_{t+1})] && (2.10) \\
&\quad + \sum_t \mathbb{E}_{\tilde{P}_{k(t)}}[\tilde{u}_t(s')] - \mathbb{E}_{P_{k(t)}}[\tilde{u}_t(s')] && (2.11) \\
&\quad + \sum_t \tilde{r}(s_t, a_t) - \bar{r}(s_t, a_t) && (2.12) \\
&\quad + \sqrt{\frac{5}{2} T \log \frac{8T}{\delta}}\,.
\end{aligned}
$$

Term 2.8 is the optimism term, since we choose our MDP and policy to maximise $\tilde{\rho}$, $\tilde{\rho}$ will be higher than ρ^* so this term is negative. Terms 2.9 and 2.10 are the off-policy error, as they sum to $\sum_t \mathbb{E}_{s\sim P_{k(t)}}[\tilde{u}_t(s') - \tilde{u}_t(s_t)]$. Term 2.9 we can bound by constructing a martingale difference sequence and applying Azuma's inequality (lemma 4) which gives a term in $\mathcal{O}(D\sqrt{T \log T})$. Note that term 2.10 is zero unless we are at $t = t_k$ when we recompute the extended value iteration, so we can take the sum over k instead of t and bound this quantity by a constant scaling with the number of epochs up to T. This will give a small $\mathcal{O}(DSA \log_2 T)$ contribution to the regret by lemma 1. Terms 2.11 and 2.12 are the on policy error. They are both bounded with the help of the constraints that define $\mathcal{C}$. Both will give a $\mathcal{O}_{S,A}(\sqrt{\log T})$ term times the following constant:

$$
\sum_k \sum_{(s,a)} \frac{v_k(s,a)}{\sqrt{\max\{1, N_k(s,a)\}}}\,,
$$

which we bounded above in 5 by $\mathcal{O}(\sqrt{SAT})$. It will turn out to be term 2.11 which dominates the sum.

2.3.3 Regret bound proof

Proof of theorem 8. Recall from the proof sketch that, given $\mathcal{M} \in \mathcal{C}_t$ at each step, with probability $1 - \delta/(12T^{5/4})$ we have:

$$
\begin{aligned}
R(s_1, T) &= T\rho^* - \sum_t r_t \\
&\leq \sum_t \rho^* - \bar{r}(s,a) + \sqrt{\frac{5}{2} T \log \frac{8T}{\delta}} \\
&\leq \sum_t \rho^* - \tilde{\rho}_t \\
&\quad + \sum_t \mathbb{E}_{P_{k(t)}}[\tilde{u}_{t+1}(s_{t+1})] - \tilde{u}_t(s_t) \\
&\quad + \sum_t \mathbb{E}_{P_{k(t)}}[\tilde{u}_t(s_{t+1}) - \tilde{u}_{t+1}(s_{t+1})] \\
&\quad + \sum_t \mathbb{E}_{\tilde{P}_{k(t)}}[\tilde{u}_t(s')] - \mathbb{E}_{P_{k(t)}}[\tilde{u}_t(s')] \\
&\quad + \sum_t \tilde{r}(s_t, a_t) - \bar{r}(s_t, a_t) \\
&\quad + \sqrt{\frac{5}{2} T \log \frac{8T}{\delta}} \,.
\end{aligned}
$$

Before anything else, we will add and subtract $\min_s \tilde{u}_t(s)$ twice and $\min_s \tilde{u}_{t+1}(s)$ once to the above, to rewrite this decomposition in terms of $\tilde{w}_t(s) := \tilde{u}_t(s) - \min_s \tilde{u}_t(s)$. This is an important trick in the proof, as before we were not able to bound $\|u\|$ in anyway, whereas now we have $\|w\|_\infty \leq D$ as shown above. Doing so now allows us to rewrite this as follows:

$$
\begin{aligned}
R(s_1, T) &\leq \sum_t \rho^* - \tilde{\rho}_t \\
&\quad + \sum_t \mathbb{E}_{P_{k(t)}}[\tilde{w}_{t+1}(s_{t+1})] - \tilde{w}_t(s_t) \\
&\quad + \sum_t \mathbb{E}_{P_{k(t)}}[\tilde{w}_t(s_{t+1}) - \tilde{w}_{t+1}(s_{t+1})] \\
&\quad + \sum_t \mathbb{E}_{\tilde{P}_{k(t)}}[\tilde{w}_t(s')] - \mathbb{E}_{P_{k(t)}}[\tilde{w}_t(s')] \\
&\quad + \sum_t \tilde{r}(s_t, a_t) - \bar{r}(s_t, a_t) \\
&\quad + \sqrt{\frac{5}{2} T \log \frac{8T}{\delta}} \,.
\end{aligned}
$$

We will consider these terms in sequence, starting now with the second term as we have already explained that the first one is negative. Let us denote X_t the terms of this sum. The sequence X_t is not in a helpful form, so we will begin by reordering it slightly. We define $\{Y_t\}$ a new sequence of terms defined by $Y_t = \mathbb{E}_{s'}[\tilde{w}_t(s')|\mathcal{F}_{t-1}] - \tilde{w}_t(s)$. We

only shift the first term of each X_t down one index, so we must adjust both ends of the sequence when summing them, which gives:

$$\sum_{t=1}^{T} X_t = \sum_{t=1}^{T} Y_t - \mathbb{E}_{s'}[\tilde{w}_0(s')|\mathcal{F}_0] + \tilde{w}_{T+1}(s)\,.$$

We can thus affirm $\sum_t X_t \leq \sum_t Y_t + 2D$. See that $\sum_t Y_t$ is the sum of a martingale difference sequence by lemma 4, hence by Azuma's inequality with $\epsilon = D\sqrt{5T\log(8T/\delta)/2}$ we have:

$$\sum_t X_t \leq \sqrt{\frac{5}{2}TD^2\log\frac{8T}{\delta}} + 2D\,,$$

with probability $1 - \delta/(12T^{5/4})$.

Now for term three, see that we can only have $w_{t+1} \neq w_t$ when we roll over from one epoch to the next, i.e. when $t+1 = t_k$. We can thus write the sum in terms of k:

$$\begin{aligned}\sum_t \mathbb{E}_P[\tilde{u}_t(s_{t+1}) - \tilde{u}_{t+1}(s_{t+1})] &= \sum_k^{m(T)} \mathbb{E}_P[w_k(s_{t_k}) - w_{k+1}(s_{t_k})] \\ &\leq 2\left\|w\right\|_\infty m(T) \\ &\leq 2Dm(T)\,.\end{aligned}$$

Where $m(T)$ is the number of epochs started up until T. We have upper bounded it in lemma 1, and thus obtain:

$$\leq 2DSA(2 + \log_2\frac{8T}{\delta}) + 2D\,.$$

There are two constant terms in this bound, which we'll just hide in a constant C to bound this term by $C + 2DSA\log_2 8T/\delta$.

Moving on to the fourth term the proof is mainly just algebra: expanding to vector form and manipulating summation indices and then applying Cauchy-Schwartz. See:

$$\begin{aligned}\sum_t \mathbb{E}_{\tilde{P}_{k(t)}}[\tilde{w}_t(s')] - \mathbb{E}_P[\tilde{w}_t(s')] &= \sum_k \sum_{(s,a)} \tilde{w}_t(s)\left(\tilde{P}_k - P_k\right) v_k(s,a) \\ &\leq \sum_k \sum_{(s,a)} \left\|\tilde{w}_t(s)\right\|_\infty \left\|\tilde{P}_k - P_k\right\|_1 v_k(s,a) \\ &\leq 2D\sqrt{14S\log\frac{2AT}{\delta}} \sum_k \sum_{(s,a)} \frac{v_k(s,a)}{\sqrt{\max\{1, N_k(s,a)\}}}\end{aligned}$$

Here we use lemma 5, which will get rid of the last term. We thus finish up the proof:

$$\begin{aligned}&\leq 2D\sqrt{14S\log\frac{2AT}{\delta}}(\sqrt{2}+1)\sqrt{SAT} \\ &\leq 2DS(\sqrt{2}+1)\sqrt{14AT\log\frac{2AT}{\delta}}\,.\end{aligned}$$

There is only one final term remaining now. This term only involves the estimated and real reward distributions, so under the assumption that the true rewards are in the confidence set $\mathcal{C}$ we can bound it easily:

$$\begin{aligned}\sum_t \tilde{r}_t(s_t,a_t) - \bar{r}(s_t,a_t) &\leq \sum_{(s,a)}\sum_k \|\tilde{r}_k(s,a) - \bar{r}(s,a)\|_\infty v_k(s,a) \\ &\leq \sqrt{\frac{7}{2}\log\frac{2SAT}{\delta}} \sum_{(s,a)}\sum_k \frac{v_k(s,a)}{\sqrt{\max\{1,N_k(s,a)\}}}\,.\end{aligned}$$

Note that the norm is bounded by the definition of $\mathcal{C}$ by a term involving t_k which we upper-bounded each time by T, We can again apply lemma 5 to complete the bound:

$$\leq \sqrt{\frac{7}{2}SAT\log\frac{2SAT}{\delta}}\,.$$

All that is needed now is to combine all these terms:

$$\begin{aligned}R(s_1,T) \leq\ & 0 \\ &+ D\sqrt{\frac{5}{2}T\log\frac{8T}{\delta}} \\ &+ 2DSA\log_2\frac{8T}{\delta} \\ &+ 2(\sqrt{2}+1)DS\sqrt{14AT\log\frac{2AT}{\delta}} \\ &+ \sqrt{\frac{7}{2}SAT\log\frac{2SAT}{\delta}} \\ &+ \sqrt{\frac{5}{2}T\log\frac{8T}{\delta}} + C\,.\end{aligned}$$

Which we can simplify for some $c > 0$, to:

$$R(s_1,T) \leq cDS\sqrt{AT\log\frac{T}{\delta}}\,.$$

This can be made to hold with probability greater that $1 - \delta/4T^{5/4}$ by Union Bound, but we did not concern ourselves with optimising this as it isn't essential to the idea of the proof (see Jaksch et al.[17] for the rigorous details). If the reader does check out the work of Jaksch et al., the proof will look completely different, but this is mainly just presentation. The decomposition is nearly analogous but they do theirs as they go, whereas we chose to do it upfront. You would find that differences are very much superficial, but there are some quite different details which in particular give more finely tuned probability bounds. □

Algorithm 1: The UCRL2 algorithm

Input: Confidence Parameter $\delta \in (0,1)$.
Initialisation: *Set* $t = 1$, $k = 1$ *observe* s_1.
for *episode* k **do**

Epoch initialisation

$t_k = t$
for $(s,a) \in \mathcal{S} \times \mathcal{A}$ **do**

$v_k(s,a) = 0$
$N_k(s,a) = |\{\tau < t_k : s_\tau = s, a_\tau = a\}|$
$R_k(s,a) = \sum_{\tau=1}^{t_k-1} r_\tau \mathbb{I}\{s_\tau = s, a_\tau = a\}$
$\hat{r}(s,a) = \frac{R_k(s,a)}{\max\{1, N_k(s,a)\}}$
for $s' \in \mathcal{S}$ **do**

$P_k(s',s,a) = |\{\tau < t_k : s_\tau = s, a_\tau = a, s_{\tau+1} = s'\}|$
$\hat{p}(s'|s,a) = \frac{P_k(s',s,a)}{\max\{1, N_k(s,a)\}}$

Candidate set computation

Compute $\mathcal{C}$ the set of values (x,y) which satisfy the constraints:

$$\mathcal{C}^r := \prod_{(s,a)\in\mathcal{S}\times\mathcal{A}} [0,1] \cap \left\{ x : |x - \hat{r}(s,a)| \leq \sqrt{\frac{7\log(2At_k/\delta)}{2\max\{1, N_k(s,a)\}}} \right\}$$

$$\mathcal{C}^P := \prod_{(s,a)\in\mathcal{S}\times\mathcal{A}} \Delta_s \cap \left\{ y : \|y - \hat{p}(s,a)\|_1 \leq \sqrt{\frac{14S\log(2At_k/\delta)}{\max\{1, N_k(s,a)\}}} \right\}.$$

Extended value iteration

$u_0 = 0$
while $\max_s\{u_i(s) - u_{i-1}(s)\} - \min_s\{u_i(s) - u_{i-1}(s)\} \geq 1/\sqrt{t_k}$ **do**

$$u_{i+1}(s) = \max_{a\in\mathcal{A}} \left\{ \max_{r\in\mathcal{C}^r} r(s,a) + \max_{P\in\mathcal{C}^P} P \cdot u_i \right\}$$

Execution

while $v_k(s_t,a_t) < \max\{1, N_k(s_t,a_t)\}$ **do**

Take action $a_t = \pi_k(s_t)$.
$v_k(s_t,a_t) = v_k(s_t,a_t) + 1$.
Observe s_{t+1}, receive r_t.
Set $t = t + 1$

Chapter 3

Online optimisation of reserve prices

3.1 Revenue maximisation in repeated SP auctions with reserve

To apply UCRL or any almost any other state of the art RL algorithm to the problem of bidding in repeated auctions, we need the environment of these agents to define a stationary MDP. If we make the assumption that the seller is single-state then we are essentially only faced with a 1D optimisation problem. A second of careful reflection will show that if the seller is oblivious to the buyers in a repeated auction then the optimal strategy for the buyers will be to repeat the optimal behaviour from a single auction. In a second price auction this is truthful bidding, which is weakly Pareto dominant.

In the real-world scenario that interests us however we have a seller who is willing to leverage his data to improve his auction over time. Thus he is multi-state and we should hope he is Markov and stationary in order for our RL algorithms to work. We can not know what the real players in the ad-exchange business do, so we devote this section to showing that there are sensible strategies for the seller which are both stationary and Markov. This exercise outperformed our expectations and so time was dedicated to improving the resulting method so as to make it useable in practice.

3.1.1 The revenue maximisation problem

As we mentioned in the short introduction to this section, in a single second-price auction the optimal behaviour for bidders is to be truthful[42]. Optimisation of bidding strategy is one of the key areas of traditional auction theory, but mostly focuses on one-shot auctions. The other two areas are properties of mechanisms (*e.g.* incentive compatibility) and what interests us, revenue maximisation. These two ares are grouped together into mechanism design, a field of game theory which deals with the attribution of items in generalised auctions.

The seminal result in mechanism design which we should start this review of revenue

maximisation with is Myerson[29]. In this work on symmetric independent bidders, Myerson gives a revenue maximising mechanism known as the Myerson auction. In essence it is a second price auction with reserve, on the *virtual value* of buyers rather than their direct bids.

Definition 1. The virtual value ψ of a distribution F with density f is given for all $r \in \text{supp}(f)$ by

$$\psi(r) = r - \frac{1 - F(r)}{f(r)}\,.$$

Running the Myerson auction, and fixing the reserve of each buyer at $\psi_i^{-1}(0)$ maximises the revenue of the seller amongst all individually rational incentive compatible mechanisms.

The work of Myerson was focused on a scenario where the distributions F is a representation of the seller's uncertainty about the valuation of buyers, a form of prior if you will. In practice to actually maximise the revenue we would need F to represent the true distribution followed by the seller. So running the optimal Myerson auction can be done only if buyer's valuations are known to sellers which is never the case in practice.

Perhaps the seller can turn to approximations if he doesn't know buyer valuations? It is interesting to fall back upon Vickrey-Clarkes-Grove[42] (VCG) mechanisms and in particular to the most commonly used in practice: second price auctions. In these auctions, the seller is allowed to set a reserve price for each bidder, and the winning bidder (if any) pays the lowest price which still wins him the auction. This fixes the payment rule of a second price auction.

In a second-price auction with *personalised* reserve (unlike in a shared reserve), the *attribution* rule can be designed in different ways. In an *eager* auction the reserve is applied before bids are accepted into the auction: a bidder can take part only if he first clears his price. In other words, the winner is the highest bid amongst those that cleared their reserve. Eager second price is thus not an efficient mechanism (a bid other than the highest may win), but it generates slightly more revenue for it.

While it increases revenue the inefficiency of the eager auction means it is often passed over in favour of *lazy* second price instead. The setting of reserves in an eager auction does not help this fact: if the seller sets different reserves for different bidders, he exposes himself to criticism from buyers who may feel they are being targeted by unfair commercial practices. A lazy auction is efficient, since it only checks if the highest bidder clears his reserve, but it also generates less revenue for exactly the same reason. Lazy second price auctions with individualised reserve remain the standard in practice and in theoretical study.

Supposing the seller wants to apply an eager second-price auction, can he use some data to maximise the reserve price he uses? Unfortunately it was shown in Paes Leme

et al.[33] that computing optimal personalised reserve prices is NP-hard and even APX-hard by Roughgarden and Wang[35]. So, outside of very simple cases, it is completely unfeasible to perform an optimal eager auction.

Conveniently, however, if one fixes the *monopoly price* as the reserve for each buyer then it has been shown by Roughgarden and Wang that this eager auction generates at least one half of the Myerson auction. The monopoly price is the price optimal reserve price in a second-price auction with only one buyer. If the buyer has value distribution F it can be computed as the expectation under F of the seller's revenue:

$$r^* = \mathbb{E}_F[r\mathbb{I}\{r \leq b\}] = r(1 - F(r))\,. \tag{3.1}$$

Not only is the monopoly price optimal in the single-buyer (*posted price*) auction and a decent approximation to the optimal eager reserve but it is also the optimal personalised reserve in the lazy second-price auction. It has been shown in Paes Leme et al.[33] that lazy-second price with personalised monopoly prices as reserves itself generates at least half of the optimal eager auction.

So, in practice the optimisation of the reserve in a lazy second price is the same as the computation of the monopoly prices. It makes sense then to replace the revenue maximisation with n bidders by the parallel estimation of each of the n monopoly prices and setting them as the reserve in a lazy (or eager) second-price auction. The problem in this chapter is thus: given $b_i \sim \mathcal{B}$ can one solve

$$\max_r \mathbb{E}_{\mathcal{B}}[r\mathbb{I}\{r \leq b\}]\,. \tag{3.2}$$

Recall, that in the context of this research project, the interest is in designing Markovian stationary policies which allow a seller to adapt to bids from a bidder in a way that can be modelled as a continuous state/time Markov Decision Process so that we can study the reinforcement learning problem this model poses to the bidder. Therefore, we are looking for algorithms to solve equation 3.2 in the context of online learning. In spite of this *a priori* we will see that the online learning method we propose actually outperforms the current offline methods.

There are several problems in the way of this maximisation (eq. 3.2. In convex optimisation, when using empirical risk maximisation and generalisation theory to solve learning problems like this, we can count on some nice properties which aren't present in the revenue function in equation 3.2. $r\mathbb{I}\{r \leq b\}$ is only quasi-concave in r and since the sum of quasi-concave functions can have exponentially many minima it is doubtful that maximising the empirical revenue in this problem will lead to a meaningful solution. To resolve them we will have to consider some surrogate loss functions.

3.1.2 Some proposed surrogates

First, since we know little about this problem we want to identify the problems which prevent us from using a general non-convex solver like a sub-gradient method. Then, once this is done, we will examine some of the existing approaches in the literature.

Focusing on the instant revenue function for the seller $G(r, b) = r\mathbb{I}\{r \leq b\}$ and its associated expected revenue $\mathcal{G}(r) = \mathbb{E}_b G(r, b)$, we will find that for a wide family of distributions $\mathcal{G}$ is pseudo-concave. Pseudo-concavity relaxes concavity but also restricts strict quasi-concavity and corresponds intuitively to the functions whose gradient never points "away" from the global maximum. The reference text for pseudo-convexity and its properties is Mangasarian[21], which will be used extensively from here on out.

Definition 2 (Mangasarian[21])**.** A differentiable function f is pseudo-concave if and only if

$$f'(x)(x - y) > 0 \implies f(x) > f(y)\,.$$

Moreover if f has a unique maximum on the interior of its domain it is strictly pseudo-concave.

The standard object of study in revenue maximisation are regular distributions, since they lead to quasi-concave revenues which thus have maxima properties required by many theoretical[1] proofs. However from there it can be shown, as in proposition 11, that they are actually pseudo-concave, which is a strictly stronger statement.

Definition 3 (Myerson[29])**.** A distribution F with density f is (strictly) regular if and only if the virtual value function ψ is (increasing) non-decreasing on the support of f.

Proposition 11. *Let F with density f be (strictly) regular, then its expected revenue $\mathcal{G}$ is (strictly) pseudo-concave. Assuming f isn't fat-tailed it is maximised by values of $r \in suppf$ such that $\psi(r) = 0$.*

We must exclude fat-tailed distributions from our revenue analysis since it is easily seen that for these distribution $1 - F(x)$ decreases slower that $1/x$ so that $\mathcal{G}$ is increasing and the maximum is attained at $+\infty$, rendering optimisation both theoretically, numerically, and realistically pointless.

Proof of proposition 11. Let F with density f be regular, then $\mathcal{G}$ is differentiable with $\mathcal{G}'(r) = 1 - F(r) - rf(r) = -\psi(r)f(r)$. Denote $S = \text{supp}(f)$, for all $r \in S$ the sign of $\mathcal{G}'$ is the opposite of the sign of ψ.
Consider a $\mathcal{C}^1$ function $h : S \to \mathbb{R}$, it is pseudo-concave if and only if:

$$\forall r \in S \forall \delta > 0 \begin{cases} h'(r) > 0 \Rightarrow f(r) > f(r - \delta) \\ h'(r) < 0 \Rightarrow f(r) > f(r + \delta) \end{cases},$$

[1] Those, which establish an analytic solution, not numerical optimisation methods which require stronger concavity conditions.

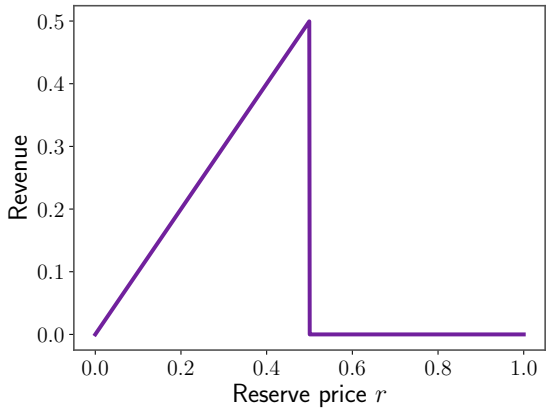


Figure 3.1: The instant revenue function $G(\cdot, b)$ for $b = .5$.

so long as $r - \delta$ and $r + \delta$ are in S. Applying this to $\mathcal{G}$, if $\psi(r) < 0$ as ψ is strictly increasing by definition 3 we have that $\mathcal{G}'$ is negative for all $y \in S$ with $y < r$ so that $\mathcal{G}$ is strictly increasing on $S \cap (-\infty, r)$. Thus for all $\delta > 0$ such that $r - \delta \in S$, we have $\mathcal{G}(r) > \mathcal{G}(r - \Delta)$.
Conversely, if $\psi(r) > 0$, we have for all $\delta > 0$ such that $r + \delta \in S$, that $\mathcal{G}(r) > \mathcal{G}(r + \Delta)$. Thus $\mathcal{G}$ is pseudo-concave. If there is a unique $r \in S$ such that $\psi(r) = 0$ (strict regularity suffices), then r is the only critical point of $\mathcal{G}$ so $\mathcal{G}$ is strictly pseudo-concave and r is its unique maximum. □

Since F has density f then it follows that $\mathcal{G}$ is at least a.e. differentiable, so it should be possible to at least compute sub-gradients for it. Since $\mathcal{G}$ is pseudo-concave these sub-gradients should[2] allow us to "flow" along the gradient to the maximum. Thus the problems mostly arises from G and not $\mathcal{G}$, let's study it more carefully.

For one fixed b, the instant revenue $G(\cdot, b)$ is given by a "saw-tooth function" (see fig. 3.1) which is only quasi-convex. It also discontinuous at b, and its limit derivatives at b are not the same on each side. The most important fact is so obvious it is easy to misss: where G is differentiable, its derivative is non-negative. However, $\mathcal{G}$ is decreasing on some interval for all bounded support distributions. For example if $F = \text{Unif}(0, 1)$ then $\mathcal{G} = x(1 - x)$ which is decreasing on $(1/2, 1)$.

The consequence of this is that given only a bid b one can compute $G(\cdot, b)$ but there is no way to go from $G(\cdot, b)$ to an unbiased gradient estimate for $\mathcal{G}$ without knowing where the maximum of $\mathcal{G}$ is, which defeats the purpose of this whole exercise. This, and not continuity or convexity, is the real problem of the stochastic revenue maximisation problem in equation 3.2. Identifying this is the first step to offering a functioning solution.

The problem of empirical revenue maximisation in auctions is thus interesting since

[2]Hand-waving away some little details here which we will focus on later.

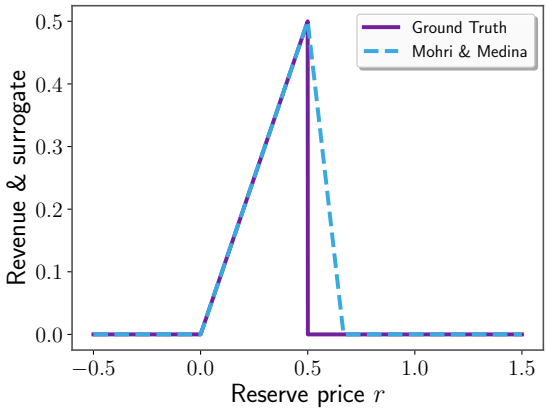


Figure 3.2: The surrogate instant revenue function of Mohri and Medina[24] for $\gamma = .3$.

its difficulty does not arise from non-convexity (since it is pseudo-concave). It arises from difficulty in estimating the gradients of the expected revenue which is what we really seek to optimise. Let's now analyse some suggested surrogate loss/revenue functions proposed in the literature.

The main method used today is the one of Mohri and Medina[24] which proposes a piece-wise linear difference of convex surrogate. Their surrogate is shown in figure 3.2. They propose this surrogate since they identified non-concavity as the main issue with optimisation of revenue. In the first part of their paper they show that designing a concave lower-bound surrogate that approximates the function well is essentially impossible, so they chose to fall back on a weaker class of functions: difference of convex (DC). Unfortunately, DC optimisation solvers are very slow. In one dimension the algorithm proposed by Mohri and Medina is essentially sorting between the $\theta(n)$ local maxima of the sum of empirical revenues. In higher dimensions (when considering predictors of bids), they must solve a Quadratic Program at each step to guarantee convergence only to a local maximum. They therefore must re-initialise this procedure to progress to a better local maximum, and repeat the operation an unknown number of times.

Another method which comes closer to ours, though by pure coincidence, it that of Rudolph et al.[36]. In this paper, the analysis of the problem by Mohri and Medina is still held and the authors propose a Bayesian approach for designing a surrogate loss which consists in an exponential smoothing and a surrogate instant revenue which is essentially the pdf of a normal placed at the bid. They come very close to our solution but are blocked in their analysis by their angle of approach. They're smoothing results of a Bayesian system, thus their operations are very different from ours and they could not provide a theoretical analysis based on convergence to a global maximiser. On the contrary, they have to rely on using the Expectation-Maximisation algorithm, which is known to only guarantee convergence to a local maximum, of which again there are *a priori* many.

3.2 A new surrogate loss

We propose to create a surrogate by making a smooth approximation of the true empirical revenue with a convolution. The nice properties of convolutions will allow us to write this also in terms of expected revenue so that we can have unbiased gradient estimates. We will be able to show also that convolution conserves all the nice properties of $\mathcal{G}$ which means we will be able to simply apply Online Gradient Ascent (OGA) to the problem. This will allow us to solve the optimisation problem efficiently, but also importantly as an online problem not an offline one.

3.2.1 Conservation of properties under convolution

In this section let us take a convolution kernel k and assume it is non-negative and normalised for simplicity. We know that revenue is also non-negative and let's assume it is normalised for simplicity too. We can do this at no cost since all the operations we will perform on G and $\mathcal{G}$ will be either integral or differential operations (expectation, convolution, differentiation) and thus multiplicative constants don't matter.

We want to show that under certain conditions on k we can conserve (strict) pseudo-concavity and other properties. We also want to show that we have added an unbiased gradient property, let's start with this since it is simple.

We denote $G_k = G * k$ and $\mathcal{G}_k = \mathcal{G} * k$ the results of the convolutions on both the instant and expected revenue. It follows immediately from Fubini's theorem that

$$\mathcal{G}_k = \mathbb{E}[G_k]\,.$$

It also follows that by the properties of convolutions

$$\frac{\partial}{\partial r}\mathcal{G}_k(r) = \mathbb{E}\left[\left(\frac{\partial}{\partial r}G(\cdot, b) * k\right)(r)\right]\,.$$

So that the gradient of $G(\cdot, b) * k$ at r is an unbiased (wrt $\mathcal{B}$) gradient estimate of $\mathcal{G}_k$ at r. Moreover, we can evaluate this gradient simply by computing $G * k'$ and evaluating it for the given b, r. Since the choice of k is up to us, we can take k at least $\mathcal{C}^1$ and such that $G * k'(r)$ has a simple closed form.

A quirk of this convolution smoothing is it has solved the worst problem with G in about two lines, but it will also take the rest of this section to show that it maintains strict pseudo-concavity which was so easily shown to hold for $\mathcal{G}$. We will rest on results from the theory of unimodal distributions which we will write in the language of pseudo-concavity, so we will give a very brief summary of the connections between the two subjects.

This primer on unimodal distributions consists of (one of) the formal definitions of unimodality in definition 4, and then of the proof of the equivalency between unimodality and pseudo-concavity in proposition 12. This will then apply and derive some properties from Ibragimov's theorem (thm. 13).

Definition 4 (Dharmadhikari and Joag-Dev[13])**.** A density f is (strictly) unimodal with mode x^* if f is (increasing) non-decreasing on $(-\infty, x^*)$ and (decreasing) non-increasing on $(x^*, +\infty)$.

Proposition 12. *A density is (strictly) pseudo-concave on its domain if and only if it is (strictly) unimodal and differentiable on it.*

Proof of proposition 12. Consider a pseudo-concave function f. Unimodality requires f to be a density, and by hypotheses f satisfies the positivity, integrability and normalisation conditions. By the properties of pseudo-concave functions[21], we have that for all $x^* \in \mathcal{S} := \operatorname{argmax}_x f$, for all $y \notin \mathcal{S}$, $f(x^*) > f(y)$ so that $f'(y)(x^* - y) > 0$. Thus f' is increasing on $(-\infty, \inf \mathcal{S})$ and decreasing on $(\sup \mathcal{S}, +\infty)$, while it is constant on $\mathcal{S}$. Thus f is unimodal.
Conversely, consider a differentiable unimodal distributions f, by definition it is non-decreasing $(-\infty, x^*)$ and non-increasing on $(x^*, +\infty)$ for some x^*. Denote $\mathcal{S} = \operatorname{argmax}_x f$, and let $x^- = \inf \mathcal{S}$ and $x^+ = \sup \mathcal{S}$, consider any $\delta > 0$, it suffices to note that $f(x^- - \delta) < f(x^-)$ and $f(x^+ + \delta) < f(x^+)$ to show that it is pseudo-concave. □

Theorists of unimodal distributions have keenly studided whether convolutions of unimodal distributions are unimodal. This is equivalent to studying whether sums of unimodal random variables (those with unimodal pdfs) are themselves unimodal. A subject of evident interest. It turns out this conjecture is incorrect in general, as is shown by the convolution of $(\delta(0) + \mathrm{Unif}(0, 1))/2$ with itself which has a mode at 0 and another at 1. Thankfully, Ibragimov proved in his 1956 note that the family of distributions which preserve unimodality by convolution (so called strongly unimodal distributions) is exactly the family of log-concave distributions and Dirac distributions.

Definition 5 (Ibragimov[16])**.** A unimodal function is strongly unimodal if its convolution with any unimodal function remains unimodal.

Theorem 13 (Ibragimov[16])**.** *A density f is strongly unimodal if and only if f is log-concave or f is a point mass.*

We ignore the point mass case, which won't help us much in terms of optimisation as $G * \delta$ is just a translation of G. Then the consequence of Ibragimov's theorem is that since we have $\mathcal{G}$ pseudo-concave, it suffices to pick k log-concave to have $\mathcal{G} * k$ pseudo-concave. It remains to show now that a unique maximum can be attained and we will have all conserved all the properties we desired and we can move on.

Proposition 14.

3.2.2 Surrogate function by convolution smoothing

Concretely, he have shown that the choice of any log-concave density k will work in the design of our surrogate losses G_k and $\mathcal{G}_k$. The obvious choice is the Gaussian, since it is the most common smoothing kernel outside of bump-functions. The standard bump function $x \mapsto exp(-1/(1-x^2))$ is also valid. In general, one should avoid log-concave functions which aren't at least once continuously differentiable ($\mathcal{C}^1$) since we would like $G * k$ to be at least $\mathcal{C}^1$. As such the uniform distributions on $[0,1]$ should be avoided, but it is known that any function can be arbitrarily approximated (in weak convergence) by a $\mathcal{C}^\infty$ one so this problem is really only superficial.

In this section we gather some example calculations and illustrations for our surrogate loss. Most of the theoretical work has been done in the previous section so it remains in this one to try and gather some qualitative insight of the effectiveness of the method and the impact of various properties of the kernel or surrogate.

Let's take a kernel k and assume $xk(x)$ is integrable on $\mathbb{R}_+$ and denote K the antiderivative of k. We can use integration by parts to write out the expression of the gradient estimate function. We use integration by parts as follows:

$$\begin{aligned} f(\cdot, b) * k'(x) &= \int_{-\infty}^{+\infty} (f(x-\tau)k'(\tau)\mathrm{d}\tau \\ &= \int_{x-b}^{x} (x-\tau)k'(\tau)\mathrm{d}\tau \\ &= x\left[k(\tau)\right]_{x-b}^{x} - \left[\tau k(\tau)\right]_{x-b}^{x} + \left[K(\tau)\right]_{x-b}^{x} \\ &= -bk(x-b) + K(x) - K(x-b)\,. \end{aligned}$$

If we take k with a well known primitive so that we have three simple components to our gradient. The gradient's behaviour will depend on the kernel, some examples can be seen on figure 3.3. Note here that only the gradient of the Gaussian smoothed surrogate is continuous since the exponential distributions is not $\mathcal{C}^1$ on $\mathbb{R}$. Figure 3.4 reveals the impact of the choice of the parameter values on the two surrogates presented in 3.3. See that in both, as the variance of the kernel goes to 0 the gradients converge (as a smooth approximation) to the ground truth.

If we can find a closed form $\bar{K}$ for the primitive of K, *i.e.* the second antiderivative of the kernel k, we can even find a closed form for the empirical smoothed revenue. Applying the same steps as above, we can obtain the formula of $G * k$ as follows.

$$\begin{aligned} f(\cdot, b) * k(x) &= x\left[K(\tau)\right]_{x-b}^{+x} - \left[\tau K(\tau)\right]_{x-b}^{x} + \left[\bar{K}(\tau)\right]_{x-b}^{x} \\ &= -bK(x-b) + \bar{K}(x) - \bar{K}(x-b)\,. \end{aligned}$$

This confirms our result for the gradient and gives a nicely smoothed version of G as can be seen in figure 3.5.

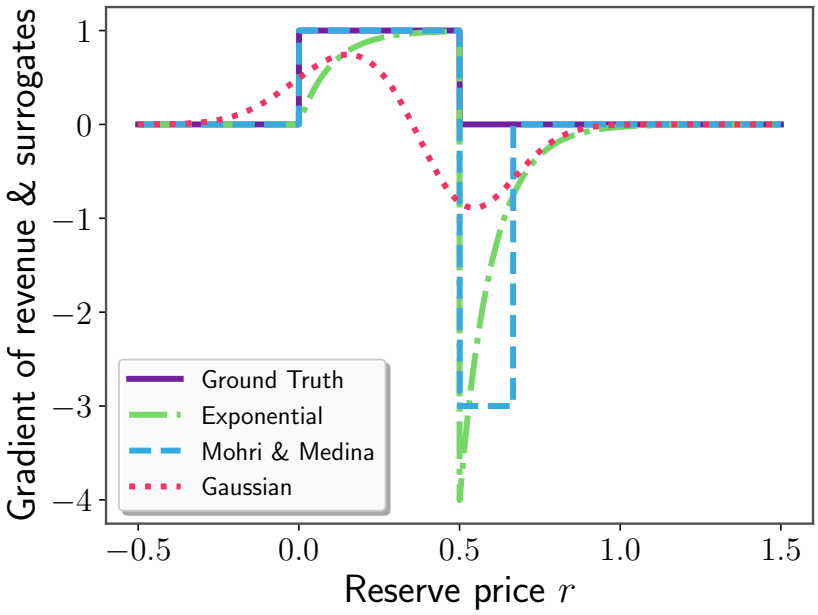
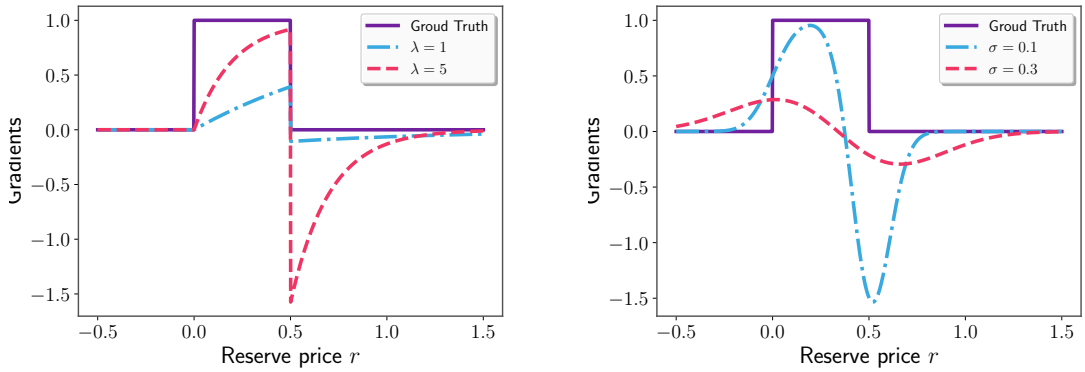


Figure 3.3: Gradients of surrogate revenue functions.

The reader will note the differences the choice of a kernel makes. A kernel defined $\mathbb{R}_+$ will lead to a revenue surrogate that is zero on $\mathbb{R}_-$. If it is not continuously differentiable on $\mathbb{R}$ this will also cause issues, such as the maximum not necessarily being a point of zero gradient. Both of these phenomena can be observed with the exponential kernel $x \mapsto \lambda \exp(-\lambda x)$ This will complicate things so we encourage the use of the Gaussian kernel and we will use it throughout this paper. However, in general, so long as the assumptions given are satisfied, the results will hold for all kernels. Using the exponential kernel is entirely possible, but it is a peculiar choice.

So far we have focused on the empirical revenue curve since it is the one which caused the problems in the analysis of the problem. Let's now turn to the expected revenue and make sure that all is well when using our surrogate.

It is easily seen on figure 3.6 that different surrogates lead to different behaviour but also that this behaviour can vary greatly based on the bid distribution. This is an argument in favour of focusing on the qualities of the empirical surrogate. However, we not in this figure that there seems to be several good reasons to pick the Gaussian kernel to design our surrogate. Note that when the bid distribution is highly skewed, non-symmetric surrogates lead to some rather unpleasant expected revenues. The one from Mohri and Medina [24] gives a monopoly price that is so close to 1 in figure 3.6b that it will generate almost no revenue. In fact, this surrogate always over-estimates the monopoly price since it is a global upper bound to G. The exponential has such a steep gradient in its empirical revenue that it risks higher numerical instability when the distributions is not a well-behaved one like on figure 3.6a.



(a) Exponential kernel. **(b)** Gaussian kernel.

Figure 3.4: Impact of parameters of different kernels on the gradients.

In conclusion, the Gaussian stands out as the stable choice of surrogate amongst those we tried (*i.e.* those with closed forms). It has the advantage of distributions the mass of G over all of $\mathbb{R}$, which tends to ease the skew and tail behaviour of distributions. It creates nice tails which will allow an optimisation algorithm to glide efficiently towards the maximum. And finally, since it is a two-parameter distribution we can directly control its mean and variance independently from each other. Since as $\sigma^2 \to 0$, $k(x) \to \delta_0(x)$, by tuning the variance we can reduce the smoothing and thus we will be able to trade off precision of the surrogate against ease of optimisation.

3.3 Optimisation of the surrogate

Now that it has been properly defined, it is time to tackle the optimisation of the surrogate revenue function. As a reminder we have two objectives in mind in this exercise: first is to create a sequential optimisation procedure which will converge to a maximum of the surrogate, and second is for it to be markovian and stationary so that we can turn it into an MDP for RL agents to compete against. Let's start by going over the precise results about gradient descent that we will use and then we will look at the particular case of the surrogate.

3.3.1 OGA/SGA - reminders

We assume that the reader is familiar with at least the basics Stochastic Gradient Descent (SGD), so we will keep this overview technical. We consider descent in this section because it is the classical framework but we will switch back to ascent in the next section. The first thing we should note is that we use SGD and Online Gradient Descent (OGD) in-

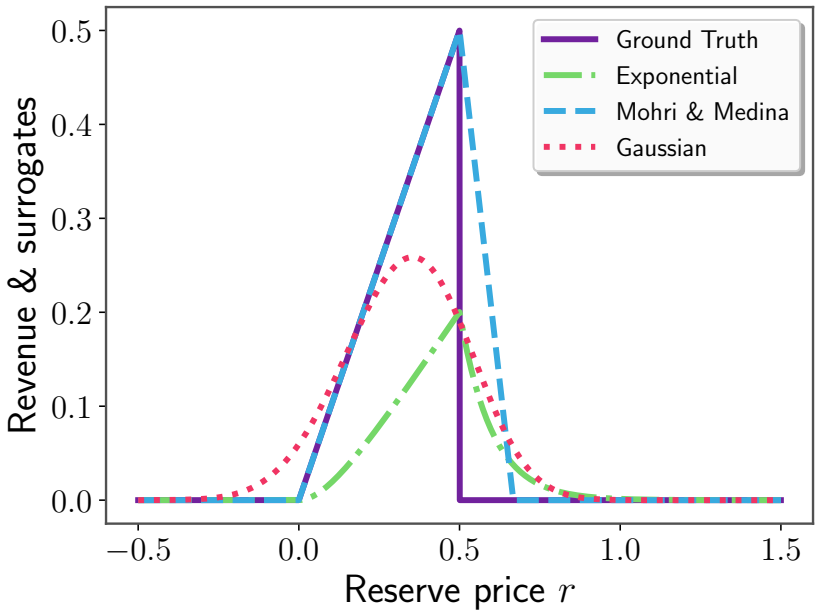


Figure 3.5: Empirical revenue surrogate functions.

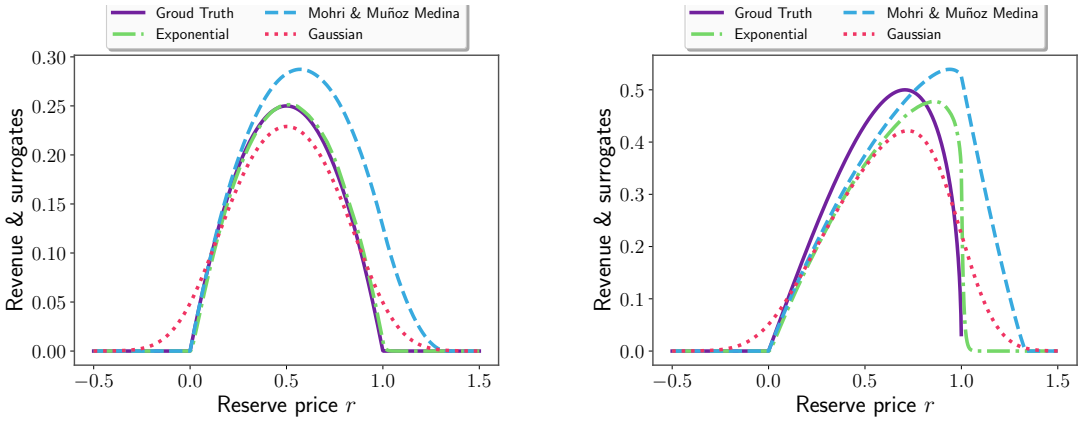
terchangeably: this only holds when the samples from SGD arrive sequentially by repeated sampling, it is not the same framework as repeated epochs on a single dataset. Likewise this also considers that the OGD is stochastic rather than observing real gradients.

This section contains three results which we will compare and draw from: all three are "convergence" proofs of OGD. They differ by their minute details which make the three proof methods quite disctinct. The first demonstrates the trick of Polyak averaging (also known as averaged gradient descent), and is a simple classical proof. However it gives only convergence in expectation of the average iterate, whereas to be rigorous we would want a stronger form of convergence, preferably almost sure (a.s.) convergence. The other two will adress this problem as well as relax its assumptions.

Proposition 15. *Let $L(x, w)$ be convex in x with $\mathcal{L} := \mathbb{E}_w[L(w, x)]$ and assume that gradient estimates g^t satisfy $\mathbb{E}_w[\|g(w^t, x)\|_2^2] \leq B^2$ for all t. Assume $\|w\|_2 \leq r$, and let $\alpha_t = D/(B\sqrt{2t})$, the iterates of projected subgradient descent satisfy:*

$$\mathbb{E}[\mathcal{L}(\bar{x}_T)] - \max_x \mathcal{L} \leq \frac{3B}{D\sqrt{T}}\,.$$

Proposition 15 is the most frequently given statement of stochastic gradient descent's convergence. However it is felt that it is neither the most powerful or the most insightful. This method essentially hides the importance of the variance of the gradients by squashing it under the averaging operation. It is more interesting to note the behaviour of the



(a) Uniform distribution on $(0, 1)$.

(b) K^a distribution $a = 2$, $b = 10$.

[a] This distributions has CDF which is the inverse of a Kumaraswamy(a, b) distribution.

Figure 3.6: Expected surrogate revenues under two bid distributions.

algorithm when the step sizes of the gradient are of the same magnitude as their variance and study the stability of the process in this stationary domain like Bottou does.

We chose to present two different proofs of almost sure convergence of gradient descent since they take somewhat different paths, and make quite different assumptions. We begin by presenting a proof of almost sure convergence of online gradient descent, in proposition 16. We must precede it with the definition of the particular kind of learning rate sequence which is used in gradient descent.

Definition 6 (Robbins-Monro sequence)**.** A positive sequence $\{a_t\}_t$ is a Robbins-Monro sequence if it satisfies to Robbins-Monro conditions:

$$\sum_{t=1}^{\infty} a_t = \infty \qquad \& \qquad \sum_{t=1}^{\infty} a_t^2 < \infty \,.$$

Proposition 16 (Almost sure convergence of SGD (1) [9, sec 4])**.** *Let a function f satisfy the following:*

1. *$\mathcal{L}$ has a unique minimum x^* to which its gradient points :*

$$\inf_{\|x-x^*\|_2^2>\epsilon} (x - x^*)\nabla_x \mathcal{L}(x) > 0.$$

2. *The updates $g(x, w)$ are unbiased estimators of the gradient : $\mathbb{E}_w[g(x, w)] = \nabla_x \mathcal{L}(x)$.*

3. *The updates have affine variance : $\mathbb{E}_w[g(x, w)^2] \leq A + B(x - x^*)^2$ for some $A, B \geq 0$.*

Let $\{\alpha_t\}_t$ be a sequence of positive learning rates which satisfies the Robbins-Monro conditions (definition 6). Online stochastic gradient descent with learning rates $\{\alpha_t\}_t$ converges almost surely to x^* for any initialisation.

With proposition 16, Bottou shows that the assumptions one would intuitively think of for gradient descent to work are in fact quite sufficient. Assumption 1 indicates that the objective needs to have a gradient which always points in the direction of the minimum so that following the gradient leads to the optimum. This property is of course verified by convex functions but convexity is not a necessary assumption. In fact there are many non convex functions which verify this property. It very closely resembles the definition of pseudo-convexity, and this is in fact quite a necessary condition for assumption 1.

However, it also requires that for all $x \neq x^*$ there are no sequences of points $\{x_n\} \to x$ such that $\nabla\mathcal{L}(x_n)(x_n - x^*) \to 0$. This means that there are no parts of the function where the gradient is asymptotically close to 0 but does not cross 0 implying a minimum. If this were the case, OGD approaching this point could converge to it because of its decreasing step size. Obviously, if a strictly pseudo-convex function is $\mathcal{C}^2$ then this condition is fulfilled and assumption 1 holds. Therefore it comes relatively free since we can pick k a $\mathcal{C}^2$ kernel to fulfil the condition on $\mathcal{G} * k$ regardless of $\mathcal{G}$'s differentiability.

Assumption 1 is all that needs to be known about the function for convergence, which is remarkably simple and exactly what one would expect intuitively as we stated in section 3.1.2. The next two conditions then concern the stochastic nature of the gradient estimates. The first requires that the gradient estimates be unbiased, which can in general be adapted to sub-gradients if needed. The second is a control on the variance of the estimates. By requiring the variance of the gradient estimates the grow at most linearly as we move away from the minimum we can insure that the gradient steps of the function will be strong enough to move towards the optimum and not be drowned out by noise. If the variance can grow too fast then a bad initialisation might lead to infinite dithering behaviour inside a region very far from the optimum.

The final (implicit) condition is the technical constraint of using a Robbins-Monro sequence of learning rates. This condition always appears in gradient descent, but the proof that is done by Bottou particularly highlights which steps it is required for. The proof of proposition 16 has the following structure, outlined by Bottou[9]: we will study the distance to the optimum over time (a Lyapunov function), and to begin we will give two lemmas (17 and 18) that are sufficient conditions for convergence of this distance. Next we will set up the problem so we can apply the lemmas and finally we will prove that this distance must converge to 0. Let's begin with lemma 17. It states that given a point x^* and a sequence of points x^t, the distance $d_t = (x^* - x^t)^2$ as a sequence (a.k.a. a Lyapunov process) converges if the sum of its increasing parts converges.

Lemma 17. *Let $\{d_t\}_t$ be a positive sequence. If $\sum_{t=1}^{\infty}(d_{t+1} - d_t)_+$, the sum of positive differences of d converges, then the sequence $\{d_t\}_t$ converges.*

Proof of lemma 17. Let $D_t^+ = \sum_{t=1}^{\infty}(d_{t+1} - d_t)_+$, and $D_t^- = \sum_{t=1}^{\infty}(d_{t+1} - d_t)_-$ be the series of positive and negative variations of $\{d_t\}_t$. By hypothesis, D_t^+ converges to D^+ so that now, for all t, $0 \leq d_t = d_1 + D_t^+ + D_t^- \leq d_1 + D^+ + D_t^-$ and thus $D_t^- \geq -d_1 - D^+$. This means the negative variations $\{D_t^-\}_t$ is a lower bounded decreasing sequence so it converges to some D^-. Since $d_t = d_1 + D_t^+ + D_t^-$, this implies the convergence of d_t to $d_1 + D^+ + D^-$. □

One can view lemma 17 as implying that dampened (decreasing variance) positive sequences converge to a limit as a sort of converse of bounded monotonic ones. It is a straightforward lemma which can be applied to online or batch (non-stochastic) gradient descent. However, to prove proposition 16 we will need a generalisation of lemma 17 to stochastic objects. In particular we will extend it to measurable quasi-martingales in lemma 18.

Lemma 18. *Consider the natural filtration $\mathbb{F} = \{\mathfrak{F}_t\}_t$ of a positive random sequence $\{d_t\}_t$, and the indicator function $\delta_t := \delta(e_t)$ where e_t is the event $\{\mathbb{E}[d_{t+1}|\mathfrak{F}_t] > d_t\}$. If $\sum_{t=1}^{\infty}\mathbb{E}[\delta_t(d_{t+1} - d_t)]$ converges, then $\{d_t\}_t$ is a quasi-martingale and d_t converges almost surely to $d \geq 0$.*

This is in spirit very much the same result as the previous one, but it can be applied to Lyapunov processes where the iterates are random. Let's now begin the proof of proposition 16 by setting up the problem to apply lemma 18.

Proof of proposition 16. We begin by defining a Lyapunov process of random variables $\{d_t\}_t$ with $d_t := \|x^t - x^*\|_2^2$ the squared Euclidian distance of the iterate x^t of OSGD to the optimum x^*. By definition of the gradient descent update, for random samples $w_t \in \mathbb{W}$ from W, we have

$$\begin{aligned} d_{t+1} = \left\|x^{t+1} - x^*\right\|^2 &= \left\|x^t - x^* - \alpha_t g(x^t, w_t)\right\|^2 \\ &= \left\|x^t - x^*\right\|^2 - 2\alpha_t g(x^t, X_t)(x^t - x^*) + \alpha_t^2 \left\|g(x^t, w_t)\right\|^2 . \end{aligned}$$

Taking the conditional expectation on X_t and rearranging yields:

$$\begin{aligned} \mathbb{E}[d_{t+1} - d_t|\mathfrak{F}_t] &= -2\alpha_t \mathbb{E}[g(x^t, X_t)](x^t - x^*) + \alpha_t^2 \mathbb{E}_W[g(x^t, w_t)^2] \\ \mathbb{E}[d_{t+1} - (1 + \alpha_t^2 B)d_t] &\leq -2\alpha_t \nabla\mathcal{L}(x^t)(x^t - x^*) + \alpha_t^2 A . \end{aligned} \tag{3.3}$$

Let's define a re-normalisation sequence $\{\eta_t\}_t$ where $\eta_t = \Pi_{i=1}^{t-1}(1+\alpha_t^2 B)^{-1}$ which converges to $\eta > 0$. This gives, dropping the negative term:

$$\mathbb{E}[\eta_{t+1} d_{t+1} - \eta_t d_t] \leq \alpha_t^2 \eta_t A .$$

We are almost ready to apply lemma 18. Define $d'_t := \eta_t d_t$, we can lower bound the expectation with the sum of positive variations:

$$0 \leq \mathbb{E}[\delta_t(d'_{t+1} - d'_t)] = \mathbb{E}[\delta_t \mathbb{E}[d'_{t+1} - d'_t|\mathfrak{F}_t]] \leq \mathbb{E}[d'_{t+1} - d'_t] .$$

Since $\sum \alpha_t^2 \eta_t A$ converges, by lemma 18 we have $d_t \overset{\text{a.s.}}{\to} d \geq 0$. So far, we have proven that the sequence $\{d_t\}_t$ of distances to the optimum (known as a Lyapunov process) converges to a finite quantity $d \geq 0$ under the assumptions of the proposition. What remains now is to prove that the event $d > 0$ has probability 0 so that $d = 0$ almost surely. To do so, let's return to equation 3.3, which now implies, by convergence of $\sum_t d_t$, that

$$0 \leq \sum_{t=1}^{\infty} \alpha_t (x^t - x^*) \nabla \mathcal{L}(x^t)$$

converges as well. Assume for a contradiction that there is some event with non-zero probability which leads to $d > 0$. Under this event, we must have that $(x^t - x^*)\nabla \mathcal{L}(x) \geq \epsilon > 0$ for all $t \geq N$ for some N. But by the first Robbins-Monro condition on $\{\alpha_t\}_t$ this causes the sum to diverge, which is in contradiction with the convergence we showed just above. Therefore, with probability 1, $d_t \overset{\text{a.s.}}{\to} 0$, which means OGD converges the optimum. □

The interesting part of proposition 16 is that it shows with a very simple proof that the hypothesis of SGD are quite weak, and that stability arguments require the Robbins-Monro property to create a a Lyapunov process which converges to 0. One can also see that dealing seriously with the randomness of gradients leads to martingale arguments. Unfortunately, This proposition does not give a convergence speed, just that convergence will happen with probability 1. In Appendix 3.5 we present a proof which gives an asymptotic speed for convex functions, and in Bach and Moulines[5] a result is given for strongly-convex functions.

As a reminder, our objective is to build an online gradient ascent algorithm which will be able to solve the revenue optimisation problem. On a low level it is an efficient, well studied, simple and elegant solution to a large category of online maximisation problem. On a high level, it is both a Markov (for constant learning rate) and a stationary dynamic, which corresponds to what we wanted, and it now remains to check that it solves the problem of revenue maximisation in our surrogate revenue.

3.3.2 Application to our surrogate

The next section concerns the application of online stochastic gradient ascent to our surrogate loss function. After all, this method is concerned with solving the online optimisation problem, which is in fact why we highlighted several proof methods. When we presented our smooth surrogate function,

Recall that by proposition 14, if the bid distribution $\mathcal{B}$ is strictly regular, which is the setting in which we know[29] how to optimise reserves if $\mathcal{B}$ is known, then convolution of the expected revenue with a log-concave probability distribution will yield a strictly

pseudo-concave function. So $\mathcal{G}_k$ has a unique maximum and is pseudo-concave if k is log-concave, which holds *e.g.* for a Gaussian kernel.

This condition is a good step on the way to fulfilling the conditions of 16. It implies that for k log-normal the inequality holds for all $x \neq x^*$. For the inequality to hold in the infimum however we need a stronger condition. Specifically, we need to forbid the following scenario: there is $x \neq x^*$ such that some sequence of points $x^t \to x$ can have gradients which go to zero. If this were to happen, we could descend along the gradient generating the sequence of iterates x^t and converge to x instead of x^*. In terms of the condition this means the infimum is 0 even though all points satisfy the condition of pseudo-convexity that $(x - x^t)\nabla_x \mathcal{L}(x) > 0$.

A careful inspection of the above scenario will reveal that if $\mathcal{L}$ is strictly pseudo-concave, this can only happen in $1D$ if $\mathcal{L}'(x^t) \to 0$ as $x^t \to x$ while $\mathcal{L}'(x) \neq 0$. This is implies that the derivative of $\mathcal{L}$ is discontinuous at x. Thus is suffices to avoid it require that $\mathcal{L}$ is $\mathcal{C}^1$. If k is $\mathcal{C}^1$ on $\mathbb{R}$, then so is $\mathcal{G}_k$, which means that it suffices to satisfy assumption *1* in proposition 16 that k is $\mathcal{C}^1$ and log-concave, like a Gaussian function.

Can we satisfy the two other conditions? We have shown in section 3.2.1 that assumption *2* can be easily satisfied by taking the gradient $G * k'(r_t, b_t)$ at time t as a direct consequence of using smoothing by convolution. The third condition can be bounded rather crudely with $B = 0$, as follows in proposition 19.

Proposition 19. *Under our assumptions the gradient of $G * k$ is bounded by*

$$\mathbb{E}_{\mathcal{B}}[g(x,b)^2] \leq \sup k^2 \, \mathbb{E}_{\mathcal{B}}[b^2] \, .$$

Proof of proposition 19.

$$\begin{aligned} g(x) = G * k'(x) &= \int_{-\infty}^{+\infty} \tau \mathbb{I}\{0 \leq \tau \leq b\} k'(x-\tau) \mathrm{d}\tau \\ &= \int_0^b \tau k'(x-\tau) \mathrm{d}\tau \\ &= [\tau k(x-\tau)]_0^b - \int_0^b k(x-\tau) \mathrm{d}\tau \\ &\leq bk(x-b) \leq b \sup k \, . \end{aligned}$$

Taking the expectation of the square gives:

$$\mathbb{E}[g^2(x)] \leq \sup k^2 \mathbb{E}_{\mathcal{B}}[b^2] \, .$$

□

Corollary 20. *If k is the density of a standard normal with variance σ^2, we have*

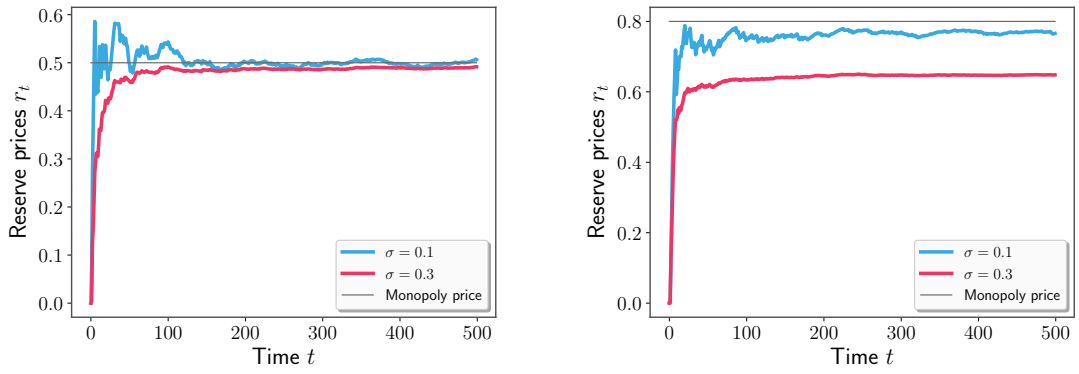
$$\mathbb{E}_{\mathcal{B}}[g(x,b)^2] \leq \frac{\sigma}{\sqrt{2\pi}} \, .$$

Thus, by proposition 19 we can verify the third condition of proposition 16. We can then affirm that if the buyer samples from a stationary distribution $\mathcal{B}$, that OGA with our smoothed surrogate converges to the global maximum of the expected surrogate revenue. Theorem 21 takes note of this fact.

Theorem 21. *Against any stationary buyer, OGA with learning rate $\eta_t \propto t^{-1}$ a.s. converges to the maximiser of the expected surrogate revenue $\mathcal{G}_k$.*

Unlike the state-of-the-art surrogate in Mohri and Medina [24], our family of surrogates converges with a very simple algorithm (OGA). Moreover, their surrogate has potentially exponentially many local maxima, while our surrogate has only one: its global maximum. Our method thus provides a triple improvement: fully online updates, very cheap update steps, and convergence to a global maximum.

Let's continue with the two example distributions above and give some examples of the convergence of the algorithm for various surrogates designed using our method. Figures 3.7 and 3.8 give two sides of the same coin: the former is a typical single trajectory and later is simply the averaged trajectory (computed by Monte-Carlo). On figure 3.7 the reader can evaluates the "variance" or "noisiness" of the descent, while on figure 3.8 the reader can view the expectation of the descent trajectory.



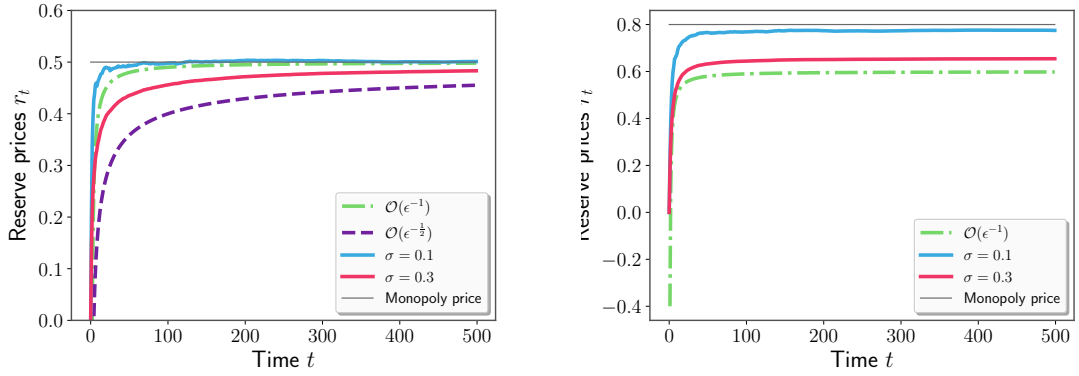
(a) Uniform distribution on $(0, 1)$. **(b)** K distribution $a = 2$, $b = 10$.

Figure 3.7: Single trajectory of OGA for two Gaussian surrogates.

First observation: Convergence of the descent is fast, and its speed is very dependent on the variance of the Gaussian kernel chosen (all other factors are held constant in this experiment). This dependence can be explained by the shape of the surrogate expected revenue which may become smoother and/or more concave for larger variances of σ^2 as it can be observed on figure 3.6b.

This same effect is also responsible for the large difference in noise in the iterates seen between both curves on figures 3.7a and 3.7b. A larger variance can almost remove the stochastic nature of the gradient estimates by smoothing out the function so heavily,

which leads to the almost noiseless descent for $\sigma = .3$. However, 3.7b demonstrates that this increased stability comes at the price of a higher displacement of the maximum (from ~ 0.8 to ~ 0.6 on figure 3.7b). This displacement of the maximum is heavier the more the revenue curve is skewed, see that for a symmetric revenue and a symmetric revenue, the mass of the revenue can't be redistributed too unevenly, but if the distribution is strongly skewed the transport may be significant. No attempts were made to analyse this effect quantitatively in this project due to time contraints, but it is likely to yield interesting insights.

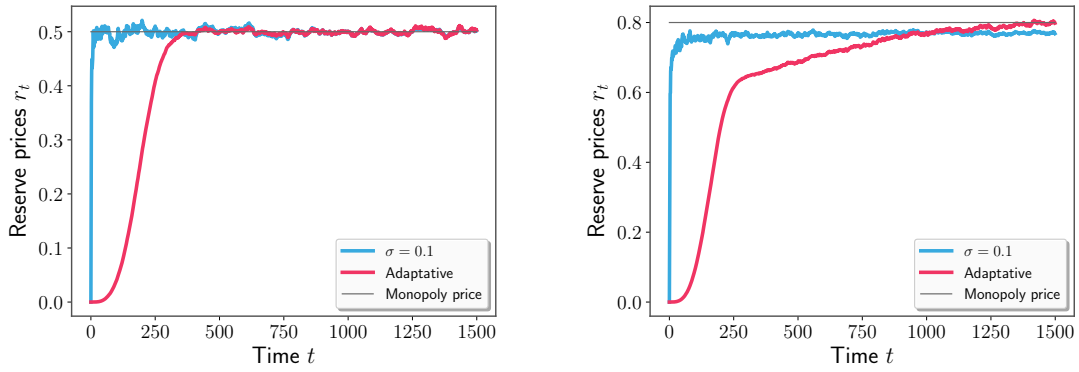


(a) Uniform distribution on $(0, 1)$. **(b)** K distribution $a = 2$, $b = 10$.

Figure 3.8: Average of 100 trajectories for two Gaussian surrogates.

In expectation now, from figure 3.8 we can make several observations. First, we can notice that in the uniform case 3.8a both surrogates do not lead to the same convergence speed. While less noisy, the convergence of $\sigma = 0.3$ is distinctly of the order of $\mathcal{O}(1/\sqrt{\epsilon})$, unlike the convergence of $\sigma = 0.1$ which is $\mathcal{O}(\epsilon^{-1})$. This may explain some of the theoretical difficulties we encountered in our theoretical analysis of the descent speed. The first curve has the speed known for the descent of a pseudo-concave objective function, but the second is much faster than any proven result to our knowledge. The interesting observation is to note that in figure 3.8b however, both curves follow $\mathcal{O}(\epsilon^{-1})$, which indicates that this speed up is a result of the interaction between the strength of the smoothing of the kernel and the shape of the true revenue.

Another perk of gradient descent is that the seller can begin his learning with a surrogate which very loosely approximates the true loss, and improve it over time. For this, let the gradient ascent converge to a value, then restart the algorithm with a lower variance from the current value. This sharpens the approximation of the true revenue around maximum iteratively while maintaining the required properties of the surrogate. In our experiments we also kept the current value of the learning rate when resetting the algorithm, which leads to the smooth transitions of figure 3.9. In figure 3.9a, the time taken to find the optimal σ, which is around 0.1 is quite fast, at around 300 steps and the learning phase is characterised by very rapid improvements which taper off once the domain of the optimal kernel is reached.



(a) Uniform distribution on $(0, 1)$. **(b)** K distribution $a = 2$, $b = 10$.

Figure 3.9: Average of 100 trajectories for Gaussian surrogates with constant versus decreasing (adaptative) variance. Learning rates of both curves are equal at $t = 1000$.

In this experiment, we set $\sigma = 100/t$, which shows that even the crudest of estimation methods are still very competitive when using an adaptive variance. Figure 3.9b even shows a behaviour where after an initial learning phase the improvements are linear until they reach peak performance and stabilise. In practice this method can be used to tune σ by following improvements of the revenue and cutting off the decrease once the revenue no longer increases. It seems a plausible extension to design a fully adaptive procedure which would not only decrease σ but adapt to changes in revenue, up or down.

A rigorous treatment of this optimisation procedure now requires us to bound how much revenue is expected to be lost when using our surrogate, for different values of the bandwidth of the smoothing kernel. We will need some form of smoothness on the expected revenue, such as Lipschitz continuity. If not, consider a revenue which is extremely "peaked" at its maximum: even if the maximisers of the surrogate and the true revenue are very close, the gap between the revenue generated by both could be arbitrarily large. We thus assume that $\mathcal{G}$ is L-Lipschitz.

Under the assumption of Lipschitz continuity, proposition 22 gives a uniform bound on the gap between the two functions. Of course, if k is the pdf of a probability distribution K then the integral simplifies to $\mathbb{E}_K[|X|]$. For example, in the case of the Gaussian kernel this value is $\sqrt{2/\pi\sigma^2}$.

Proposition 22. *For all integrable kernels k, we have:*

$$|\mathcal{G}_k(x) - \mathcal{G}(x)| \leq L \int |\tau|\, k(\tau) \mathrm{d}\tau \,.$$

Proof.

$$\begin{aligned}
|\mathcal{G}_k(x) - \mathcal{G}(x)| &= \left| \int_{-\infty}^{+\infty} \mathcal{G}(x-\tau)k(\tau)\mathrm{d}\tau - \mathcal{G}(x) \right| \\
&= \left| \int_{-\infty}^{+\infty} (\mathcal{G}(x-\tau) - \mathcal{G}(x))k(\tau)\mathrm{d}\tau \right| \\
&\leq \int_{-\infty}^{+\infty} L\,|x-\tau-x|\,k(\tau)\mathrm{d}\tau \\
&= L\int_{-\infty}^{+\infty} |\tau|\,k(\tau)\mathrm{d}\tau\,.
\end{aligned}$$

□

From this global bound on the point-wise difference between the functions, we can derive a bound on the lost revenue at the optimum of the surrogate. This is given by proposition 23.

Proposition 23. *For all k strongly unimodal, denote r^* and r_k^* the maximisers of $\mathcal{G}$ and $\mathcal{G}_k$ respectively, we have:*

$$\mathcal{G}(r^*) - \mathcal{G}(r_k^*) \leq 2L\int_{-\infty}^{+\infty} |\tau|\,k(\tau)\mathrm{d}\tau\,.$$

Proof.

$$\begin{aligned}
\mathcal{G}(r^*) - \mathcal{G}(\tilde{r}^*) &\leq |\mathcal{G}(r^*) - \mathcal{G}_k(\tilde{r}^*)| + |\mathcal{G}_k(\tilde{r}^*) - \mathcal{G}(\tilde{r}^*)| \\
&\leq \mathcal{G}(r^*) - \mathcal{G}_k(r^*) + L\int_{-\infty}^{+\infty} |\tau|\,k(\tau)\mathrm{d}\tau \\
&\leq 2L\int_{-\infty}^{+\infty} |\tau|\,k(\tau)\mathrm{d}\tau\,.
\end{aligned}$$

□

In summary, using our surrogate we have cheap convergence to the surrogate's maximum and we can control the lost revenue as a functional of the kernel chosen. So far we considered only the bid distribution, but if we assume now that bidders are truthful and bid precisely the valuation they see in each object, then corollary 24 quantifies precisely the asymptotic gap between our method (surrogate plus OGA) and the optimal second price auction an all-knowing seller could possibly run.

Corollary 24. *If the bidder is truthful then the auction run with our method asymptotically loses at most $2L\int|\tau|\,k(\tau)\mathrm{d}\tau$ in expected revenue compared to the optimal second price auction.*

3.3.3 Tracking of monopoly prices

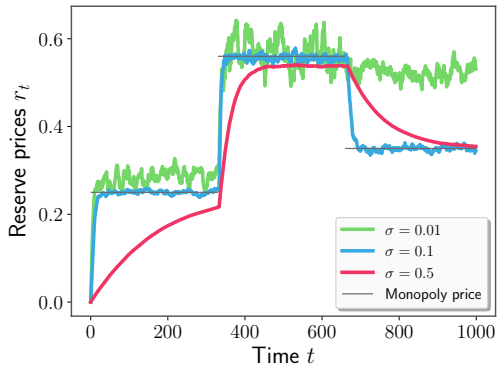
So far, the experiments and theoretical results in this chapter have numerically shown that we have devised an interesting method for revenue maximisation, that functions well (sometimes better than expected even). So it is clearly of independent interest, but we originally set out to devise a stationary markovian update to reserve prices which could be used as a basis to formulate an RL problem in continuous space and time. So far, we have an update which is 1-Markov in the reserve price, but it its dependence on time is a problem. Fixing this problem only requires us to take a constant learning rate, such an ascent could be made continuous time by simply shifting to gradient flow (the continuous time equivalent of gradient ascent).

There is actually a very good motivation for a seller to fix a constant learning rate on his own which can be understood thanks to the developments we have detailed in this section. We assumed bidders were stationary, but as soon as we have a bidder which updates its strategy with RL this will no longer be the case. Shouldn't the seller then prefer to forgo the last few percents of approximation to the reserve price in favour of the ability to adapt to changing bid distributions? This can be seen as almost being analogue to using an eternal exploration strategy versus an explore-then-commit one in a bandit.

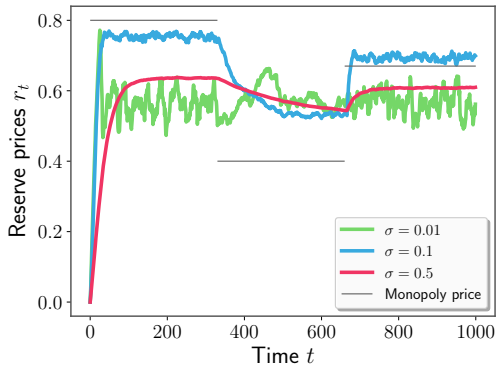
Let's say the seller fixes the learning rate η, then the first question is if he can effectively track relatively infrequent switches in the bid distribution. Figure 3.10 show that he can, but with several caveats. One, the impact of the smoothing on the speed and noise of convergence becomes even more important. See that on each phase the blue curve ($\sigma = 0.1$) will accumulate linear regret (due to noise), but it always performs much much better than the red one ($\sigma = 0.5$) because it converges much faster so it can respond to switches much more efficiently. Two, if the smoothing is too weak ($\sigma = 0.001$ on figure 3.10a) then it will end up suffering from the same problems as the original empirical revenue, and will not be able to decrease when responding to a switch to a lower monopoly price. This problem also concerns the decreasing learning rate case but it is much graver here.

In this constant learning rate case, there are two parameters for the seller to tune: variance, and learning rate. Both have similar effects on performance and should in practice be optimised by grid-search or some such heuristic. For now we will forget about switches and study only one phase with a constant learning rate. On figures 3.11 and 3.12 the reader can see that with a constant learning rate and smoothing, one converges into a "band" around the maximum of the surrogate. The width of this band needs to be minimised where possible while maintaining a good convergence speed to avoid the problem with the red curve in figure 3.10a.

The reader can see on figure 3.11 that the width of the band decreases with the learning rate, which makes sense since this reduces the size of each step. However, it is clear that a too small learning rate leads to much slower convergence, see the red curve with

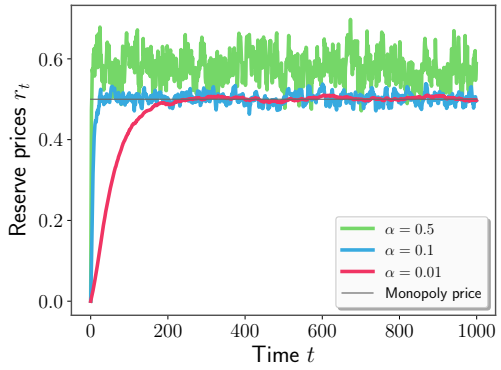


(a) Uniform distributions on $(0, 0.5), (0.5, 1), (0, 0.7)$.

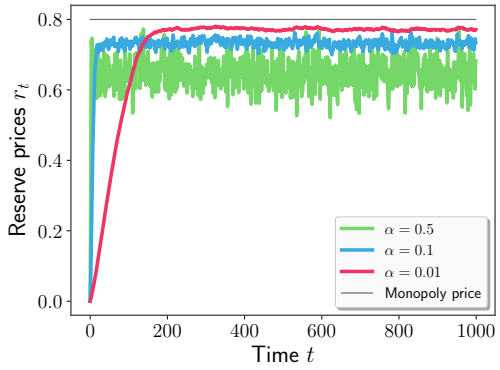


(b) K distributions $(2, 10), (10, 2), (1, 3)$.

Figure 3.10: Tracking of switches in bid distributions with different constant learning rates.



(a) Uniform distribution on $(0, 1)$.



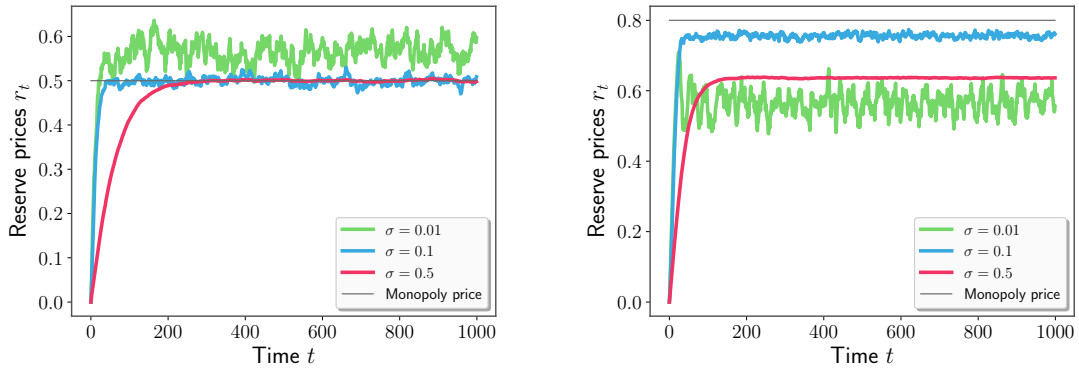
(b) Kumaraswamy distribution $a = 2$, $b = 10$.

Figure 3.11: Average of 100 trajectories for the Gaussian surrogate with $\sigma^2 = 0.1$ for different constant learning rates.

$\alpha = 0.01$ (remember this descent is on $[0, 1]$) which is what we want to avoid. Conversely, a learning rate too high leads to high variance and a very wide band which means a lot of lost revenue. In the middle is a sweet spot where convergence is almost as fast as the green curve but the noise is almost as low as the red curve, which the blue curve illustrates. 0.1 was often a good choice for our experiments on $[0, 1]$ but fine-tuning will be necessary for any production grade implementations.

Conversely to the learning rate, the same behaviour is observed with varying values of σ. A value too large leads to slower learning rates, as was observed in figure 3.8a above, which we want to avoid. Less smooth[3] surrogates however lead to higher variance as the value of the bid has a higher impact on the size of the step. This is exemplified by the

[3]In the Lipschtiz sense, all are $\mathcal{C}^\infty$.



(a) Uniform distribution on $(0, 1)$. **(b)** K distribution $a = 2$, $b = 10$.

Figure 3.12: Average of 100 trajectories for three Gaussian surrogates with different variances.

green curves; a compromise exists again in the blue curve. While the impact of the kernel is more subtle than the learning rate (note the shifts of the maximum in figure 3.12b) they both have the same essential behaviour. Both need to be tuned empirically, and most likely jointly, hence the use of grid-searches, or perhaps even a descent algorithm of its own, or some adaptation of our adaptive procedure from figure 3.9.

There is a good theoretical, empirical, and rational grounding behind the use of constant learning rates in gradient ascent for the maximisation of revenues in lazy second-price auctions. This environment can be used to do reinforcement learning. It is 1-Markov in the state since we have eliminated the dependence in time from the learning rate, it is valid as a transition kernel on any sub-interval on $\mathbb{R}$ once wishes to restrict the study to by using the projected gradient ascent method. This chapter is thus completed and its goal has been attained, we can write discrete and continuous in time updates based on the prior choice by the seller of a kernel, a learning rate, and a variance. Unfortunately, the original objective of this chapter brought us to study a problem (monopoly price learning) which was much more difficult than expected and thus there is no time left to study the interaction between UCRL2 and a discretised gradient descent or to develop the study of the control problem in this repport.

Conclusion

The objective of this research project was to study the problem of online learning in auctions, focusing on the consequences of the interactions of multiple ML agents in the context of repeated auction. In repeated auctions that Criteo takes part in, all processes are automated and thus quantitative parameters of auctions (reserve prices, shading, etc.) are computed by learning agents. This poses new problems which we studied in this paper.

3.4 Summary of results

On the seller side, we sought to design an online algorithm to learn the monopoly price since we saw that it was a reasonable trade-off between learnability and revenue maximisation. In mind the problem of multiple ML algorithms interacting, we also sought to design a tracking algorithm which could learn the monopoly price of a bid distribution quite well and be able to adapt if the underlying distributions changes. In order to provide a toy model to study the buyer problem we also wanted this algorithm to be Markovian and stationary.

In chapter 3 we presented the work that has been done on the learning of monopoly prices. We analysed the problem and the curent theory around it, and we showed why current methods fail to implement simple online learning algorithms. We solved the unbiased gradients problem by proposing the use of a surrogate loss function based on convolution with a smooth log-concave kernel. This surrogate can be used with classical online gradient ascent, or any other first order algorithm, but can also be used with a constant learning rate to track non-stationary buyers. At the same time, it can also be updated online so as to reduce the bias and asymptotically learn the true monopoly price instead of the maximiser of the surrogate.

On the buyer side, our long-term goal was to study extensions of reinforcement learning to continuous time, state, and action spaces. To do this we first wanted to draw up an analysis of the problem of online RL in unknown environments to determine what the difficulties would be in continuous problems and to devise some insights into some promising research directions. We thus sought to study the discrete time state of the art (UCRL2) and adapt it to continuous time.

We gave an analysis of UCRL2, including breaking it down into the three sub-tasks of online RL: learning, planning, and the balancing of the explore-exploit dilemma. We analysed extended value iteration (EVI) as the key contribution of the algorithm which

allows for optimistic planning to be done in one step in a meta-MDP. Computation of EVI requires the use of a trick which is specific to finite state MDPs but it seems like adapting it to continuous time actions and states would be a promissing direction of future research.

3.5 Outlook.

The lines of research begun with this internship are not completed, both for the buyer and the seller problems. With Marc Abeille and Clement Calauzènes we are finishing an article for the AISTATS 2019 conference on our work on the online learning of monopoly prices against stationary and non-stationary buyers. This article includes some research which has not been presented in this report because it has been produced in the last few weeks, after the writing of most of this report. This includes theoretical guarantees for the width of the oscillations around the monopoly price with a constant learning rate and finite time bounds on convergence with various learning rates, adapted from the work of Moulines and Bach[28].

This paper will complete the digression we took from the buyer problem by combining all our new results for publication, but it will leave many open questions and research problems of its own. In particular our method only solves one dimensional problems, whereas it would be very beneficial to learn to incorporate features into the learning problem. In practice it is possible to create buckets, *i.e.* clusters, of the data and independently place an instance of our algorithm for each, but ideally we would like to take a parametric hypothesis class $\mathcal{H}$ and maximise the monopoly revenue for features $X \sim \mathcal{X}$ perhaps with some regularisation R:

$$h^* \in \operatorname*{argmax}_{h\in\mathcal{H}} \mathbb{E}_{b\sim\mathcal{B},X\sim\mathcal{X}}[G(h(X))] + R(h)\,.$$

Solving a maximisation of this form would allow us to exploit the public features which both parties has about the value of the item sold with any existing approximation algorithm such as a neural network. This would allow us to learn features at scale by resting on the very well studied literature from learning theory on these methods.

Unfortunately, it is not obvious that the objective in this case will be concave, or in fact even pseudo-concave so that our convolution trick with a log-concave kernel works. By using the tower rule we have:

$$\mathbb{E}_{b,X}[G(h(X))] = \mathbb{E}_X[h(X)(1 - F_{b|X}(h(X)))]\,.$$

While the term in the expectation is pseudo-concave under the (already strong assumption) that $F_{b|X}$ is regular for all $X \in \mathcal{X}$, there is no *a priori* reason that this pseudo-concavity should be preserved under the expectation. There is therefore some theoretical analysis which would need to be done to determine exactly what generalised concavity properties

are satisfied by this monopoly revenue and then determining if they will mesh well with our surrogate and gradient ascent.

We studied the feature problem but we did not manage to find any convincing results so far, and it stands as a clear short coming of our online method which demands further research. It is motivated by the large potential revenue gains, but also by the intuition that it should be possible since gradient descent extends very smoothly from one dimensional to multi-dimensional problems and that smoothing should be simple to generalise to high dimension too. Marc Abeille and I intend to leave this line of work for now and focus on the buyer problem.

In the next few months I will be starting a CIFRE Ph.D. position with Marc Abeille as my industry supervisor where we will focus on the extension of stochastic control results to solve the planning problem in reinforcement learning in continuous time, state, and action. We will be working on the part of this research project (on the buyer side) which was not completed in this internship, *i.e.* designing optimal strategies to compete against a seller who is learning to set the monopoly price as the reserve in a lazy second-price auction.

Bibliography

[1] Amin, K., A. Rostamizadeh, and U. Syed
2014. Repeated contextual auctions with strategic buyers. In *Advances in Neural Information Processing Systems*, Pp. 622–630.

[2] An, M. Y.
1998. Logconcavity versus logconvexity: a complete characterization. *Journal of economic theory*, 80(2):350–369.

[3] Auer, P., N. Cesa-Bianchi, and P. Fischer
2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256.

[4] Auer, P. and R. Ortner
2007. Logarithmic online regret bounds for undiscounted reinforcement learning. In *Advances in Neural Information Processing Systems*, Pp. 49–56.

[5] Bach, F. and E. Moulines
2011. Non-asymptotic analysis of stochastic approximation algorithms for machine learning.

[6] Bartlett, P. L. and A. Tewari
2009. Regal: A regularization based algorithm for reinforcement learning in weakly communicating mdps. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, Pp. 35–42. AUAI Press.

[7] Bertsekas, D. P.
1995a. *Dynamic programming and optimal control*, volume 2. Athena scientific Belmont, MA.

[8] Bertsekas, D. P.
1995b. *Dynamic programming and optimal control*, volume 1. Athena scientific Belmont, MA.

[9] Bottou, L.
1998. Online learning and stochastic approximations. *On-line learning in neural networks*, 17(9):142.

[10] Bubeck, S., N. Cesa-Bianchi, et al.
2012. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122.

[11] Bubeck, S., N. R. Devanur, Z. Huang, and R. Niazadeh
2017. Online auctions and multi-scale online learning. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, Pp. 497–514. ACM.

[12] Cesa-Bianchi, N., C. Gentile, and Y. Mansour
2014. Regret minimization for reserve prices in second-price auctions. *IEEE Transactions on Information Theory*, 61(1):549–564.

[13] Dharmadhikari, S. and K. Joag-Dev
1988. *Unimodality, convexity, and applications*. Elsevier.

[14] Dodu, J., M. Goursat, A. Hertz, J. Quadrat, and M. Viot
1981. Méthodes de gradient stochastique pour l'optimisation des investissements dans un réseau électrique. *EDF Bulletin de la Direction des Etudes et Recherches, série C-mathématiques, informatique*, (2):133–164.

[15] Friedlander, F. G. and M. S. Joshi
1998. *Introduction to the Theory of Distributions*. Cambridge University Press.

[16] Ibragimov, I. A.
1956. On the composition of unimodal distributions. *Theory of Probability & Its Applications*, 1(2):255–260.

[17] Jaksch, T., R. Ortner, and P. Auer
2010. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(Apr):1563–1600.

[18] Krishna, V.
2009. *Auction theory*. Academic press.

[19] Lai, T. L. and H. Robbins
1985. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22.

[20] Lattimore, T. and C. Szepesvári
2018. Bandit algorithms. *preprint*.

[21] Mangasarian, O. L.
1994. *Nonlinear programming*. SIAM.

[22] McAfee, R. P. and J. McMillan
1987. Auctions and bidding. *Journal of economic literature*, 25(2):699–738.

[23] Medina, A. M. and S. Vassilvitskii
2017. Revenue optimization with approximate bid predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Pp. 1856–1864. Curran Associates Inc.

[24] Mohri, M. and A. M. Medina
2016. Learning algorithms for second-price auctions with reserve. *The Journal of Machine Learning Research*, 17(1):2632–2656.

[25] Mohri, M. and A. Munoz
2015. Revenue optimization against strategic buyers. In *Advances in Neural Information Processing Systems*, Pp. 2530–2538.

[26] Morgenstern, J. H. and T. Roughgarden
2015a. On the pseudo-dimension of nearly optimal auctions. In *Advances in Neural Information Processing Systems*, Pp. 136–144.

[27] Morgenstern, J. H. and T. Roughgarden
2015b. On the pseudo-dimension of nearly optimal auctions. In *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds., Pp. 136–144. Curran Associates, Inc.

[28] Moulines, E. and F. R. Bach
2011. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *Advances in Neural Information Processing Systems*, Pp. 451–459.

[29] Myerson, R. B.
1981. Optimal auction design. *Mathematics of operations research*, 6(1):58–73.

[30] Nedelec, T., M. Abeille, C. Calauzènes, N. E. Karoui, B. Heymann, and V. Perchet
2018. Thresholding the virtual value: a simple method to increase welfare and lower reserve prices in online auction systems. *arXiv preprint arXiv:1808.06979*.

[31] Ortner, R.
2008. Optimism in the face of uncertainty should be refutable. *Minds and Machines*, 18(4):521–526.

[32] Osband, I., D. Russo, and B. Van Roy
2013. (more) efficient reinforcement learning via posterior sampling. In *Advances in Neural Information Processing Systems*, Pp. 3003–3011.

[33] Paes Leme, R., M. Pal, and S. Vassilvitskii
2016. A field guide to personalized reserve prices. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, Pp. 1093–1102, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

[34] Puterman, M. L.
1994. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

[35] Roughgarden, T. and J. R. Wang
2016. Minimizing Regret with Multiple Reserves. In *Proceedings of the 2016 ACM Conference on Economics and Computation - EC '16*, volume 9, Pp. 601–616.

[36] Rudolph, M. R., J. G. Ellis, and D. M. Blei
2016. Objective variables for probabilistic revenue maximization in second-price auctions with reserve. In *Proceedings of the 25th International Conference on World Wide Web*, Pp. 1113–1122. International World Wide Web Conferences Steering Committee.

[37] Shen, W., S. Lahaie, and R. P. Leme
2019. Learning to clear the market. In *International Conference on Machine Learning*, Pp. 5710–5718.

[38] Shoham, Y., R. Powers, and T. Grenager
2003. Multi-agent reinforcement learning: a critical survey. *Web manuscript*.

[39] Sutton, R. S. and A. G. Barto
2018. *Reinforcement learning: An introduction*. MIT press.

[40] Tang, P. and Y. Zeng
2018. The price of prior dependence in auctions. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, Pp. 485–502. ACM.

[41] Thompson, W. R.
1933. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294.

[42] Vickrey, W.
1961. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of finance*, 16(1):8–37.

[43] Weissman, T., E. Ordentlich, G. Seroussi, S. Verdu, and M. J. Weinberger
2003. Inequalities for the l1 deviation of the empirical distribution.

Appendices

Classical Proof of Gradient descent (Polyak averaging)

This Appendix contains the omitted proof of averaged stochastic gradient descent (proposition 15). Proposition 15 is reproduced below and then a proof is given.

Proposition 25. *If we let $\alpha_t = \alpha/\sqrt{t}$ be the learning rate of OGA, given iid samples from F, and if G is L-Lipschitz and bounded in $[0, M]$. we have that the loss of the mean iterate of $\tilde{G}$ relative to the optimal value is less that $3LM/\sqrt{T}$.*

Proof. Let x^t denote the iterates, $g(x^t, X_t)$ be the gradient estimates given data point X_t we have:

$$\begin{aligned}\left\|x^{t+1} - x^*\right\|^2 &\leq \left\|x^t - x^* + \alpha_t g(x^t, X_t)\right\|^2 \\ &\leq \left\|x^t - x^*\right\|^2 + 2\alpha_t g(x^t, X_t)(x^t - x^*) + \alpha_t^2 \left\|g(x^t, X_t)\right\|^2 .\end{aligned}$$

Taking the conditional expectation on X_t yields

$$\begin{aligned}\mathbb{E}[\left\|x^{t+1} - x^*\right\|^2] &\leq \mathbb{E}[\left\|x^t - x^*\right\|^2] + 2\alpha_t(x^t - x^*)\mathbb{E}[g(x^t, X_t)] + \alpha_t^2\mathbb{E}[\left\|g(x^t, X_t)\right\|^2] \\ &\leq \mathbb{E}[\left\|x^t - x^*\right\|^2] + 2\alpha_t(x^t - x^*)f'(x^t) + \alpha_t^2 L^2 \\ &\leq \mathbb{E}[\left\|x^t - x^*\right\|^2] + 2\alpha_t(f(x^t) - f(x^*)) + \alpha_t^2 L^2 ,\end{aligned}$$

Which follows from concavity of f. By re-arranging and taking the expectation we obtain

$$\begin{aligned}f(x^*) - \mathbb{E}[f(x^t)] &\leq \frac{1}{2\alpha_t}\mathbb{E}[\left\|x^t - x^*\right\|^2] - \mathbb{E}[\left\|x^{t+1} - x^*\right\|^2] + \frac{\alpha_t L^2}{2} \\ f(x^*) - \mathbb{E}[f(\bar{x}_t)] &\leq f(x^*) - \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[f(x^t)] \\ &\leq \frac{1}{2T\alpha_1}\mathbb{E}[\left\|x^1 - x^*\right\|^2] + \frac{1}{2T}\sum_{t=1}^{T-1}\left(\frac{1}{\alpha_{t+1}} - \frac{1}{\alpha_t}\right)\mathbb{E}[\left\|x^{t+1} - x^*\right\|^2] \\ &\quad - \frac{1}{2T\alpha_{T+1}}\mathbb{E}[\left\|x^{T+1} - x^*\right\|] + \frac{L^2}{2T}\sum_{t=1}^{T}\alpha_t \\ &\leq \frac{1}{\alpha_1}M^2 + 2M^2\sum_{t=1}^{T-1}\left(\frac{1}{\alpha_{t+1}} - \frac{1}{\alpha_t}\right) + \frac{L^2}{2}\sum_{t=1}^{T}\alpha_t .\end{aligned}$$

The first sum is telescopic, hence we have

$$\begin{aligned}
&\leq \frac{2M^2}{T\alpha_T} + \frac{L^2}{2T}\sum_{t=1}^{T}\alpha_t \\
&\leq \frac{2M^2\sqrt{T}}{\alpha_0} + \frac{\alpha_0 L^2}{2T}\sum_{t=1}^{T}\frac{1}{\sqrt{t}} \\
&\leq \frac{3ML}{\sqrt{T}}\,.
\end{aligned}$$

The second to last step recalls $\alpha_t := \alpha_0/\sqrt{t}$, and the last step involves upper-bounding the sum and optimising α_0. □

Assymptotic speed and a.s. convergence

Now we propose to analyse a different proof, in proposition 26, which will show how martingale inequalities can be used to derive speed guarantees about the convergence. On the other hand, this proposition requires stronger assumptions, and doesn't have the stability considerations that make Bottou so elegant.

Proposition 26 (Almost sure convergence of SGD (2))**.** *Let $\mathcal{L} : \mathcal{X} \to \mathbb{R}$ be a proper, convex, L-Lipschitz, differentiable function on a bounded set $\mathcal{X}$ in a Hilbert space. Further assume that its minimum x^* satisfies*

$$\forall x \in \mathcal{X} \, : \, \mathcal{L}(x) - \mathcal{L}(x^*) \geq c \left\| x - x^* \right\|^2 . \tag{4}$$

Let $\{\alpha\}_t$ be a sequence that satisfies the Robbins-Monro condition, consider unbiased gradient estimates $g_t(x, w)$ with variance bounded by V. Then iterates x^t of projected SGD satisfy:

$$\left\| x^* - x^t \right\|_2 \overset{a.s.}{\to} 0 .$$

Proof. Dote $\Pi_{\mathcal{X}}$ the projection operator on to $\mathcal{X}$, and define the Lyapunov process $d^t := \|x^t - x^*\|_2^2$. Let random samples w_t be draw according to W form a sample space $\mathbb{W}$ as above. By definition

$$\begin{aligned} d^{t+1} &\leq \left\| x^{t+1} - x^* \right\|_2^2 \\ &\leq \left\| \Pi_{\mathcal{X}}(x^t - \alpha_t g_t(x^t, w_t)) - x^* \right\|_2^2 \\ &\leq \left\| x^t - \alpha_t g_t(x^t, w_t) - x^* \right\|_2^2 \end{aligned}$$

by contraction of the projection operator. Expanding gives

$$\begin{aligned} d^{t+1} &\leq \left\| x^t - x^* \right\|_2^2 - 2\alpha_t \langle g_t(x^t, w_t) | x^t - x^* \rangle + \alpha_t \left\| g_t(x^t, w_t) \right\|_2^2 \\ &\leq d^t - 2\alpha_t \langle g_t(x^t, w_t) | x^t - x^* \rangle + 2\alpha_t^2 V^2 . \end{aligned}$$

Taking the conditional expectation with respect to the natural sigma-algebra at time t, and then by convexity of $\mathcal{L}$ we have

$$\begin{aligned} \mathbb{E}[d^{t+1} | \mathfrak{F}_t] &\leq d^t - 2\alpha_t \langle \nabla_x \mathcal{L}(x^t) | x^t - x^* \rangle + 2\alpha_t^2 V^2 \\ &\leq d^t - 2\alpha_t (\mathcal{L}(x^t) - \mathcal{L}(x^*))) + 2\alpha_t^2 V^2 . \end{aligned}$$

Using condition 4 in the theorem we obtain:

$$\begin{aligned} &\leq d^t - 2\alpha_t c d^t + 2\alpha_t^2 V^2 \\ &\leq (1 - 2\alpha_t c) d^t + 2\alpha_t^2 V^2 \,. \end{aligned} \tag{5}$$

See that, by recurrence and the Robbins-Monro property, it is easily shown that $\mathbb{E}[d^t] \to 0$, but we are interested in a finer result. We are going to devise bounds on the sequence d^t, and to do so we adapt this equation in terms of a supermartingale. Let Y_n be defined as $d^n + 2V^2 \sum_{i=n}^{\infty} \alpha_i^2$. See that it is a supermartingale since $\mathbb{E}[|Y_n|] < \infty$ and by equation 5

$$\mathbb{E}[Y_{n+1} | \mathfrak{F}_n] \leq d^n + 2V^2 \sum_{i=n+1}^{\infty} \alpha_i^2 + 2\alpha_n^2 V^2 = Y_n \,.$$

It is also a sequence of positive random variables, so by the supermartingale convergence theorem $Y_n \overset{\text{a.s.}}{\to} 0$ and thus $d^n \overset{\text{a.s.}}{\to} 0$. On top of almost sure convergence we can use Kolmogorov's or Doob's inequality, or an equivalent, to obtain:

$$\mathbb{P}(\sup_{n \geq N} Y_n \geq \epsilon) \leq \frac{1}{\epsilon} \mathbb{E}[Y_N] \,.$$

For N large enough, we enter a domain where we can explicitly bound $\mathbb{E}[Y_N] = \mathbb{E}[d^N] + 2V^2 \sum_{i=N}^{\infty} \alpha_i^2$. Specifically this domain is where the size of the update α is comparable to $c\mathbb{E}[d^t]/V^2$. We give the final inequalities which rest on this fact and we will then finish the proof by proving this property:

$$\begin{aligned} \mathbb{P}(\sup_{n \geq N} Y_n \geq \epsilon) &\leq \frac{1}{\epsilon} \left(\frac{1}{\frac{c^2}{L^2} N + \frac{1}{d^0}} + \sum_{i \geq N}^{\infty} \frac{c^2 L^2}{\left(\frac{c^2}{L^2} i + \frac{1}{d^0} \right)^2} \right) \\ &\leq \frac{L^2}{c^2 \epsilon} \left(\frac{1}{N} + \sum_{i=N}^{\infty} \frac{1}{i^2} \right) \,. \end{aligned}$$

Now, see that $Y_n = d^n + \epsilon'_n$ and $\epsilon'_n \to 0$ which means that we can write this sup bound on d^n, thus d^n converges almost surely to 0.

To prove that the quantity we gave is indeed an upper bound to $\mathbb{E}[d^t]$ we return to equation 5 and to identify the domain more carefully we begin by looking at what conditions the learning rates must satisfy. We know to use a learning rate that satisfies the Robbins-Monro conditions, so let's look at something of the form $\alpha_t = D/(t + B/A)$. We want to check which values are permissible, and in particular we will verify that it holds for $A = c^2/V^2$, $B = 1/d^0$, and $D = 1/c$. Returning to equation 5 we have:

$$((t+1)A+B)\mathbb{E}[d^{t+1}] \leq (tA+B)\mathbb{E}[d^t] \left(1 + \frac{1}{t + B/A} \right) \left(1 - 2c \frac{D}{t + B/A} \right) + \frac{D^2 V^2 ((t+1)A + B)}{(t + B/A)^2} \,.$$

We proceed by recurrence, and we will prove that for all t, $(tA + B)\mathbb{E}[d^t] \leq 1$ for well chosen A, B, D. First see that for $t = 0$, this imposes $0 < B \leq 1/d^0$. Now suppose $t(A + B)\mathbb{E}[d^t] \leq 1$, for the recurrence to hold at $t + 1$ we need

$$1 - \frac{2cD - 1}{t + B/A} - \frac{2cD}{(t + B/A)^2} + \frac{D^2 V^2 ((t+1)A + B)}{(t + B/A)^2} \leq 1 \,.$$

This is equivalent to the following system:

$$\begin{cases} D^2q^2A - 2cD \leq 0 \\ B/A(D^2V^2A - 2cD + 1) - 2cD + D^2V^2A \leq 0 \end{cases} .$$

We won't solve this system in general, but we can check it is true if we take $A = c^2/V^2$ and $D = c^{-1}$. Let's now move to the asymptotic regime where $a_t \simeq c\mathbb{E}[d^t]/V^2$. This is chosen as it is the value which minimises the right hand side of equation 5. Injecting into equation 5 again would give:

$$\mathbb{E}[d^{t+1}] \leq \mathbb{E}[d^t]\left(1 - \frac{c^2}{V^2}\mathbb{E}[d^t]\right) .$$

This behaves like the differential equation (*i.e.* the continuous gradient descent) $\partial_t z = -c^2z^2/V^2$ which integrates to

$$z(t) = \frac{1}{\frac{c^2}{V^2}t + \frac{1}{d^0}} .$$

Then, if $\alpha_t := (c^2t/V^2 + 1/d^0)^{-1}$, we have as N gets large enough:

$$\mathbb{E}[d^N] \leq \frac{1}{\frac{c^2}{V^2}N + \frac{1}{d^0}} ,$$

which is what we used earlier in our inequalities. □