ONLINE MATCHMAKING: FROM CONTEXTUAL BANDITS TO LOGISTIC REGRESSION

INTRODUCTION AU DOMAINE DE RECHERCHE

Pierre Monteiller
M2 StatML
Under the supervision of Matthieu Lerasle (ENSAE) and Vianney Perchet (ENSAE)

ABSTRACT

Matchmaking is an important issue in the improvement of video game design. We investigate the modeling of such problem in an online setting and the different questions coming along. We leverage the prolific literature in contextual bandits algorithms, a similar topic, to provide baseline algorithms that we would like to outperform. Tailoring the exploration phase has a tremendous importance in the performance of any technique: we describe two algorithms with an adaptive exploration phase. Finally, we present a logistic regression based matchmaking algorithm that was developed under the idea saying that knowing the player strengths would enable us to match in a more satisfying way the players.

1 Introduction

The video gaming industry is huge and shows no signs of slowing down. While there were almost 1.82 billion video gamers across the world in 2014, this figure is steadily rising and is expected to achieve the tremendous number of 2.7 billion gamers by 2021. In terms of revenue, the global e-Sport market was valued at over one billion U.S. dollars in 2020 and it will reach almost 1.6 billion U.S. dollars in 2023. We focused in this work on a particular type of video games: those where players are playing 1VS1 matches. Designing a matchmaking system that satisfies the players is a core challenge for video games makers to increase their community of players. This crucial service that allows players to find opponents raises important issues regarding player experience. It should not lead to mismatches where strong players face weak ones, a situation which satisfies none of the players involved. Furthermore, response times can be very long: ranging up to hours of waiting until a game session can begin. Most player communities would welcome better schemes towards the elimination of malicious player behaviors and truly efficient matchmaking. This was our main motivation to study matchmaking systems and it will be our general subject in this thesis.

Moreover, in video games, players are arriving in an online fashion. This match should be as satisfying as possible, even though the system may have no initial information about the incoming player. Therefore, we chose a statistical online setting for our study, on the contrary to a batch offline statistical setting. To be more precise, information is coming one at a time rather than all in one time. Online setting has many advantages over standard learning: one should not wait for all the points to come to estimate: it is faster, points often comes in an online manner and temporality may have importance: it is more realistic and computation at time t are often only involving the incoming point: it is computationally more efficient. In online learning, the decision maker observes one point every time and refine the estimation of the parameter at every steps. To assess the quality of an online procedure or strategy, one can look carefully at several quantities: we chose to understand and dig better into the notion of regret. This quantity captures the loss the statistician endured in their estimation due to the fact of being in an online setting compared to an offline setting. Our goal would be to show the sub-linearity in time of matchmaking strategies and some nice dependencies in other variables such as dimension of this quantity.

Two main issues come along those choices of settings. The first one is that we are not interested in a parameter estimation in itself but rather in a matching policy between players. The genuine goal of our work is to develop methods and strategies that make games between players satisfying. The second one is about the trade-off between exploration and exploitation in online learning problems. At a certain time, the matchmaking knows that some games were satisfying, others no and have no clue about the rest of the possible games. The decision to adopt a certain matchmaking policy at time t is torn between exploiting previously observed games by re-matching players together and exploring new matches that can have higher satisfaction the previously observed ones.

This introduction to the subject is divided into three big parts:

- The mathematical description of our online problems and the question we will try to answer.
- Design a matchmaking policy with contextual bandits a spin off problem from multi armed bandits.
- Design a new logistic-regression based algortihm for online matchmaking.

2 Statistical Modeling of Matchmaking

We will assume that there is a pool of $[n] = \{1, ..., n\}$ players, who are queuing to be matched against one another. At each time $t = 1, ..., T$, there will be three different steps:

1. A player $I_t \in [n]$ arrives (randomly or adversarially chosen for now).
2. The system matches this player to player $J_t \in [n]$ following a certain matchmaking policy.
3. The system receives a feedback of information and update its training data set in order to improve future estimates.

Our focus will be on the rules that establish the matchmaking policy, rules that will be highly depending on the modeling choices in term of feedback information. The two information the system will have access to will be the outcome of the game and the satisfaction of that game.

2.1 Bradley-Terry Models

The outcome of a match will be assumed to follow a Bradley-Terry model [15, 6, 9, 14], which is arguably the most fundamental model in pairwise comparisons. Given a set $[n] = 1, ..., n$ of n players, let $\Gamma = \{\gamma \in \mathbb{R}_+^n |\ \sum \gamma_i = 1\}$ be the set of model parameters. θ_i will represent the strength of player i. In a Bradley-Terry model, given a pair of team i and j, the probability that team i beats team j, under the parameter $\gamma \in \Gamma$ is defined as follows $\mathbb{P}(i \text{ beats } j)|\ \Gamma) = \frac{\gamma_i}{\gamma_i+\gamma_j}$. The main issue in this definition of such model is that this function $f_{i,j} : \gamma \mapsto \frac{\gamma_i}{\gamma_i+\gamma_j}$ is not convex with respect to γ. More conveniently, we can rewrite the previous equation in order to work with a well convex function, the logistic function. Let $\theta_i = \ln \gamma_i - \ln \gamma_1$, then $\gamma_i = \frac{\exp^{\theta_i}}{\sum_{j=1}^n \exp^{\theta_j}}$. This is a bijection between Γ and $\Theta = \{\theta \in \mathbb{R}^n |\ \theta_1 = 0\}$. The probability expressed in terms of γ may be rewritten in terms of θ such as follows:

$$\mathbb{P}(i \text{ beats } j)|\ \Gamma) = \frac{\gamma_i}{\gamma_i + \gamma_j} = \frac{1}{1 + \exp^{-(\theta_i - \theta_j)}} = \mathbb{P}(i \text{ beats } j)|\ \Theta)$$

We recognize the sigmoid function, function appearing in the logistic regression. However with such reparametrization, there is a major drawback. It is easily understandable that the the spread of the different forces of the players will have a tremendous consequence on the performance of our algorithms. We define the following quantity capturing the spread of the true value of the forces θ

$$K = \max_{i,j \in [n]} \frac{\gamma_i}{\gamma_j} = \max_{i,j \in [n]} \exp^{\theta_i - \theta_j}$$

Note that $\ln K = \max_{i,j \in [n]} |\theta_i - \theta_j| \leq 2||\theta||$ and $||\theta|| = O(\sqrt{n \ln K})$. Generally, K and $||\theta||$ are infinitely large. For example, consider the case where $n = 2$ and $\gamma = (1, 0)$(player 1 beats player 2 almost surely). The corresponding θ in the new domain is $\theta = (0, -\infty)$. So, $K = ||\theta|| = +\infty$. Infinitely large domain is not desirable especially for online learning, which typically requires knowledge of the diameter of the domain $||\theta||$. Optimization in Bradley-Terry models often assumes that there is no "too strong team" for which $K = \infty$. This is mainly to avoid pathological cases in the existence of the maximum likelihood estimator. To alleviate this burden we regularize our function. Thanks to that regularization, the MLE will exist and we chose that way to deal with that issue of $K = \infty$.

A crucial parameter in BT models is the strengths of the players. The distribution of θ will play a tremendous importance in the following, as being bounded is one of most common assumptions on the incoming data. Therefore we have two choices for the modeling of θ: Either θ is supposed to lie in a bounded subset of $\mathbb{R}^n$ such as $[-b, b]^n$. θ is then deterministic, Either θ is drawn from a particular distribution, and is therefore stochastic.

The first assumption makes things easier to handle, however it is very restrictive: no players can be very beginners nor experts in the game. Thus, we will look carefully at the second option. To model the natural distribution of strengths for the players, one could easily think about a Gaussian distribution or anything that has a similar spread. This is pretty natural as we might think of a game in which most of the players are of average level, being very good requires a lot of

training and being very bad at this game may be easily avoided by training a bit. However, most of the games does not have this kind of profile in the forces of the player. Consider a game in which the more you play the better you are, no matter your intrinsic ability to play. Then the tail of the distribution will be heavier than in the case of a sub Gaussian tail. That is the case where it is the case when the training has a strong correlation with the force of a player. Therefore, it is very rare to be super bad at a game and the forces of good players is much more spread than in the previous ideal case of Gaussian situation. There is a fundamental asymmetry in this case that can be much more realistic compared to the ideal symmetric Gaussian case. The main concern of these two types of distribution is the fact that they are not bounded. However they are with high probability. We investigate the two cases of sub Gaussian and sub Gamma distributions.

2.2 Satisfaction measure

The satisfaction measure should be symmetric in a first approximation: both players should be satisfied evenly for a game. Also, we made the assumption of optimistic and positive thinking players: the satisfaction should always be positive. Therefore, we chose unsurprisingly a positive definite kernel function as a measure of satisfaction.

Definition 1. A function $K : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ is a positive definite kernel function if for all $x \in \mathbb{R}^n$ the matrix $(K(x_i, x_j))_{i,j}$ is symmetric definite positive.

A natural candidate for such Kernel function is the Gaussian Kernel, with a parameter σ that catches this tolerance described above. The parameter σ will describe the tolerance of the players to play against slightly better or worse, the bigger σ, the less tolerant will be the players.

2.3 Regret Minimization

The goal of the statistician, and by transitivity of the video game designers, is to maximize the satisfaction of the players. The performance of an online learning algorithm is measured by the cumulative loss it suffers along its run on a sequence of observed events. The observed events will be written as $(x_t, y_t)_{t=1,...,T}$, where $x_t = (0, .., 1, ..., -1, ..., 0)^T$ with 1 placed at i and -1 put at j; y_t will be equal to 1 if i wins, -1 otherwise (no game will be considered as a tie game). The cumulative satisfaction is $\sum_{t=1}^{T} K(\theta_{I_t}, \theta_{J_t})$. There is no strategy (sequence of J_t) that correctly predicts the correct answers of all observed instances. In this case, we would like the cumulative satisfaction of our online algorithm not to be too lower by much the cumulative satisfaction of the best strategy. Formally, we assess the performance of the learner using the notion of *regret*, term coming from the field of "online convex programming". This term was introduced by this paper [11] in the beginning of the second half of the XXth century, and recently developed in [17] but this setting was introduced some years earlier by Gordon in [10]. Given any fixed strategy, we define the regret of an online learning algorithm as the deficit of satisfaction for not consistently predicting with the strategy $J = (J_t)$ responding to incoming sequence of players (I_t),

$$R_T(J) = \max_{(J_t^*)_{t=1,\dots,n}} \sum_{t=1}^{T} K(\theta_{I_t}, \theta_{J_t^*}) - \sum_{t=1}^{T} K(\theta_{I_t}, \theta_{J_t}) \tag{1}$$

The main goal in the project is to control this quantity as sharply as possible, in terms of lower and upper bounds. Regret is expected to be sub-linear, and it is even better if it is dominated by $O(\sqrt{T})$ or $O(\ln T)$. The regret and its control should depend on T, n the number of players and D the spread of forces. The objective of this thesis is to describe several techniques, methods and algorithms that achieve some bounds on the regret. After describing our method, we will develop our own algorithm based on logistic regression and we hope to prove better lower bounds that existing ones.

3 Contextual Bandits for Matchmaking

The contextual Bandits seems to be very close to our problem of finding the best matchmaking policy. In contextual bandits, one observes a context and should pull the arm with the highest reward, reward depending on the the context. One is trying to find the best policy $\pi \in \Pi = [n]^{n-1}$ between contexts and arms, that is to say, if we note $\mathcal{X}$ the space of contexts and $[n]$ the arms, the best map in terms of regret $\pi : \mathcal{X} \longmapsto \{1, \dots, n\}$. In general contextual bandits setting, there is no link between contexts and arms. A common application of contextual bandits is to improve user satisfaction by tailoring recommendations to specific user's needs (for example, news articles that the user is likely to be interested in) [13]. However, the incoming player I_t can be seen as the context and our matchmaking problem may now be cast as a contextual bandits problem.

3.0.1 Exp-3 Algorithm

Compared to a typical multi-armed bandit problem, the only aspect the contextual bandit problem adds is that the learner receives a context at the beginning of the round, before it needs to select its action. When we have a finite context set, which is our case here, one simple approach is to use a separate off-the-shelf bandit algorithm for each context, such as Exp3 (Exponential-weight algorithm for Exploration and Exploitation) [7]. The main idea is to update for each player I, a probability distribution p_I over the set of the others players $[n] \setminus \{I\}$ capturing the propensity of player I to play with another. The system will choose at time t the player J_t according to the distribution p_{I_t}. To take advantage from some convex properties of the satisfaction, we will update the probability distributions with exponential weights,

Theorem 1. The regret of the Exp3 algorithm for matchmaking satisfies

$$R_T \leq \sqrt{2Tn^2 \ln n} \tag{2}$$

The algorithm Exp3 has then a regret in $O(\sqrt{Tn^2 \ln n}) = O(\sqrt{Tn \ln |\Pi|})$, which is promising as it is sub-linear and has a small dependency in terms of number of players. Moreover, Exp-3 is often computationally feasible to run. It has a linear computational complexity. However, this algorithm was taken directly off the shelf and is not tailor for the particular problem of contextual bandits. Therefore, this algorithm will serve at first in our benchmark as a very efficient algorithm in terms of regret but very inefficient in terms of computational complexity. Its main advantage is that is very close to common algorithms existing in multi-armed bandits literature.

3.1 Epsilon-greedy method

An obvious approach (called the the ε-greedy algorithm) is to have a preliminary exploration phase (for $\tau = \varepsilon T$ trials – which is a fraction of the total number of trials), and reap its benefits in a pure exploitation phase. To fairly evaluate the performance achieved by the pure exploration in τ trials, we restrict the algorithm to learn only from the exploration phase rewards, and not from the exploitation phase rewards, for this naive case. Note that we assume the knowledge of the time horizon T here. The strategy has then those two phases

1. Exploration phase: when I_t arrives, the system chose J_t uniformly at random
2. Exploitation phase: after this exploration, one should determine the matchmaking policy π between players which maps players to the one who has the biggest observed satisfaction r and choose J_t according to it. This policy will be i.e.

$$\pi_\tau \in \arg\max_{\pi} \sum_{t=1}^{\tau} r_t \mathbb{1}_{\pi(I_t)=J_t} \tag{3}$$

   where r_t is drawn from a certain distribution $\mathcal{P}(\cdot|I_t, J_t)$ with mean $K(\theta_{I_t}, \theta_{J_t})$.

Formally, the process is stated in the following Algorithm 1. To assess the quality of a matchmaking policy, we define its value as the expected average reward by choosing π:

$$V(\pi) = \mathbb{E}_{I_t} \mathbb{E}_{r_t \sim \mathcal{P}(\cdot|I_t, J_t)}[r_t|I_t] \tag{4}$$

The regret will then be rewritten as: $R_T = T \max_\pi V(\pi) - \sum_{t=1}^{T} V(\pi_t) = TV(\pi^*) - \sum_{t=1}^{T} V(\pi_t)$.

Theorem 2. For a well picked value of τ, the regret of ε-Greedy is at most of order $O(T^{2/3})$.

The ε-Greedy algorithm has then a bigger regret in $O(T^{2/3})$ than Exp-3 but has a much easier computational implementation. The only computationally demanding step is when the system has to find the best policy after the exploration phase, this can be done naively by a linear search for each player which has a logarithmic complexity (some better solution should exist). Therefore, to the contrary to the Exp3 algorithm, this algorithm will serve at first in our benchmark as a very efficient algorithm in terms of implementation but inefficient in terms of regret. Its main advantage is its simplicity in terms of analysis.

3.2 Tailoring Exploration in Contextual Bandits for Matchmaking

The ε-greedy method chooses players uniformly at random during the exploration phase, and therefore incur a sub-optimal $O(T^{2/3})$ regret. We should be able to improve on this if we use our previously observed rewards to influence our exploration strategy. For example, if a policy has poor empirical regret on our existing observations, and the policy's

Algorithm 1 ε-Greedy for Matchmaking

Input: Training Set $S = \emptyset$, parameter τ for explorqtion phase

for $t = 1, \ldots, \tau$ **do**
  Observe I_t
  Select $J_t \in [n] \setminus \{I_t\}$ uniformly at random
  Observe the reward r_t
end for
Determine the optimal policy

$$\pi_\tau \in \arg\max_{\pi} \sum_{t=1}^{\tau} r_t \mathbb{1}_{\pi(I_t)=J_t}$$

for $t = \tau + 1, \ldots, T$ **do**
  Observe I_t
  Select $J_t = \pi_\tau(I_t)$
  Observe the reward r_t
end for

empirical regret is an accurate estimate of its true regret, we should stop exploring that policy. Ultimately, we would like to achieve the optimal $O(\sqrt{T})$ regret of Exp3 without its computational complexity linear in $|\Pi|$.

Recently, the PolicyElimination [8] and *Importance-weighted Low-Variance Epoch-Timed Oraclized CONtextual BANDITS* (Ilovetoconbanditxs) algorithms were developed [1] , which uses an adaptive exploration strategy to achieve the statistically-optimal $O(\sqrt{T})$ regret, while still being computationally feasible to run. In order to avoid a $\Omega(|\Pi|)$ time complexity, we restrict ourselves to algorithms that only access the policy class through an optimization oracle, which returns the policy with the highest empirical reward given the observations so far. The algorithm can maintain a distribution over policies, but it must be sparse (i.e. most policies must have zero probability) in order to avoid computing probabilities for each policy. The following discussion is based on those two papers. We only present the first algorithm.

3.3 Notations

Recall that we use $V(\pi)$ to denote the expected reward of a policy. We define the empirical estimate of matchmaking policy π value using t samples as

$$\hat{V}_t(\pi) = \frac{1}{t} \sum_{s=1}^{t} r_s \frac{\mathbb{1}_{J_s=\pi(I_s)}}{p_t}$$

We will use $Q \in \Delta(\Pi)$ to denote distributions over $\Pi = [n]^{n-1}$, that is $Q(\pi) \geq 0$ and $\sum Q(\pi) = 1$. Given an incoming player i, we will also abuse notation and use $Q(j|i)$ to be the induced distribution over players, given the incoming player. That is,

$$Q(j|i) = \sum_{\pi\,:\pi(i)=j} Q(\pi)$$

In other words, $Q(j|i)$ is the sum of the probabilities of all policies which recommended player j a for an incoming player i. Finally, since we want to ensure that our reward estimates have low variance, we should prevent the probability of any action from being too small. To accomplish this, we assign a minimum probability for each action γ. Then, there is $(1 - n\gamma)$ probability left, which we split among the arms according to the $Q(j|i)$ distribution. We use the shorthand $Q^\gamma(j|i)$ to be the smoothed mixture of Q with the uniform distribution over players.

3.4 PolicyElimination Algorithm

PolicyElimination algorithm [8], is an iterative elimination based algorithm. At each iteration, the algorithm evaluates every surviving policy according to the IPS estimator on the data collected so far. Policies which have a low empirical regret according to these estimates are retained, while the rest are eliminated. The algorithm then constructs a probability distribution for exploration at the next round over the surviving policies.

Algorithm 2 PolicyELimination Matchmaking

Input: T, failure probability δ

Initialize $\Pi_0 = \Pi$

Define $\varepsilon_t = 2\sqrt{\frac{2n\ln(n^{n-1}T/\delta)}{t}}$ and $\gamma = \min\{\frac{1}{2n}, \sqrt{\frac{\ln(n^{n-1}T/\delta)}{2nT}}\}$

for $t = 1, \dots, T$ **do**

  Observe I_t

  Find Q_t distribution over Π_{t-1} such that

$$\forall \pi \in \Pi_{t-1}, \quad \mathbb{E}_{I_t}\left[\frac{1}{Q^\gamma(\pi(I_t)|I_t)}\right] \leq 2n$$

  Select $J_t \in [n] \setminus \{I_t\}$ accordingly to $Q_t^\gamma(j|I_t)$

  Observe the reward r_t

  Update $\Pi_t = \{\pi \in \Pi_{t-1},\ \max_{\pi'} \hat{V}_t(\pi') - \hat{V}_t(\pi)\}$

end for

The philosophy behind this algorithm is to have distribution over matchings, that from the observed results tries to determined whether or not a policy is efficient or not. Good matchings, i.e. the ones with high values, are then candidate for the selection of player J_t. The player J_t is drawn with high probability if a lot of good policies are sending incoming player I_t on J_t, and the better the policy is, i.e. policy with high $Q(\pi)$ probability to be selected, the bigger impact it has on the selection of J_t.

We look carefully at the regret control. Similarly as done before for the ε-Greedy algorithm, one should examine $\hat{V}_t(\pi) - V(\pi)$ and argue that this quantity is small for all feasible policy $\pi \in \Pi_t$. We would like to use some concentration inequalities as before. However, the probability $p_t = Q^\gamma(J_t|I_t)$ is heavily depending on the past making $\{(I_t, J_t, p_t, r_t)\}$ not i.i.d. random variables on the contrary to the situation of ε-Greedy algorithm. The random variables $Y_t(\pi) = r_t \frac{\mathbb{1}_{J_t=\pi(I_t)}}{p_t} - V(\pi)$ defined such that

$$\hat{V}_t(\pi) - V(\pi) = \frac{1}{t}\sum_{s=1}^{t} Y_s(\pi)$$

are not i.i.d. anymore. However, this sequence of variables forms a martingale difference sequence with respect to the canonical filtration $\mathcal{F}_t = \sigma((Y_s(\pi))_{s=1}^{t-1})$, that is to say a sequence of zero mean variables such that $\mathbb{E}[Y_t(\pi)|Y_1(\pi), \dots, Y_{t-1}(\pi)] = 0$. Now, we have at our disposal some Bernstein-like concentration inequality for martingale difference sequence described in Lemma 1. For a proof of that concentration result see [4].

Lemma 1. Let $Y_1, \dots, Y_n$ be a martingale sequence difference with $Y_i \leq R$ for all i. Then, with $V_n = \sum_{i=1}^n Var_n(Y_i)$, for any $\lambda \in [0, 1/R]$ and $\delta \in (0, 1)$, we have with probability $1 - \delta$

$$\sum_{i=1}^{n} Y_i \leq (e-2)\lambda V_n + \frac{\ln 1/\delta}{\lambda} \tag{5}$$

Theorem 3. The regret of PolicyElimination algorithm is bounded with probability $1 - 2\delta$ such that

$$R_T \leq 17\sqrt{2Tn\ln\frac{n^{n-1}T}{\delta}}$$

The main computational bottlenecks of PolicyElimination are the elimination step and the requirement on the expectation of the inverse of distribution Q_t^γ. More precisely, we saw in the proof of theorem **??**, that this constraint has mainly consequences on the variance of $Y_t(\pi)$, on the variance of instantaneous regrets. We wonder then if we can avoid the computational costly step of elimination altogether, enforce the variance constraints over all the policies and somehow encourage the distribution Q_t to not place mass over policies with a large empirical regret. Actually, it is impossible to enforce the variance constraints for all $\pi \in \Pi$, and also obtain low regret. It is intuitive since variance constraint is costly in terms of information, hence in terms of regret, and we would like to avoid it for bad policies.

4 Online Matchmaking and Logistic Regression

The true goal of my master's thesis is to develop an algorithm that outperforms the baseline algorithms. Our starting point was that the contextual bandits algorithms never used the underlying structure of the Bradley-Terry models, thus never used the information that we have from the outcomes of the different games. Therefore, we tried to leverage this information and the parametric formulation of Bradley-Terry models. This led us to study online logistic regression in a particular setting. This subsection is dedicated to present all the results of our matchmaking algorithm we gathered, even if they are partial. We first investigate online learning and Bradley-Terry models, and it turns out that the literature is not dense. We focused at the beginning on [15], where Matsumoto studies regret of Follow-the-Leader procedure in Bradley-Terry models. However, the setting was purely adversarial and the system had no influence on the choice of player J_T at time t. We then decided to study existing methods of ratings in sports, especially the paper of Kiraly and al [12]. But, it seems that these ratings are inconsistent with the strength of the player. Finally, we decided to focus on a heuristic based on the likelihood of the model that enables to estimate the vector of strength. This subsection begins with the design of the objective function, combining the statistical likelihood of our observations and the satisfaction we want to maximize. Then, we present the gradient descent algorithm trying to minimize that objective function, and describe the criterion of choice for J_t in terms of exploration-exploitation trade-off. Finally, we present a first bound on the regret based on the convergence in online logistic regression.

4.1 Designing an objective function

There is an underlying tension in the system between exploitation and exploration, between satisfying the players and estimating the forces of these latter. Therefore we should play on that fact to build a new update rules that will catch this tension, which characterizes our system. We decide to build an algorithm that captures this tension. Our idea was that a method based on the optimization of the likelihood function will enforce exploration while maximizing the satisfaction will tend to enforce exploitation.

The statistical model we are dealing with is very similar to a logistic regression, as the outputs are Bernoulli realizations and the probability to have a positive result is given by the Bradley Terry model. To be more precise, we will write in the future σ the sigmoid function, i.e. $\sigma(x) = \frac{1}{1+e^{-x}}$. An observation of a game between players i and j at a time $t \in \mathbb{N}^*$ will be encoded as a pair $(x_t, y) \in \mathbb{R}^n \times \{-1, 1\}$ with $x_t = (0, .., 1, ..., -1, ..., 0)^T$ with 1 placed at i and -1 put at j; y will be equal to 1 if i wins, -1 otherwise (no game will be considered as a tie game). On the other hand, the global satisfaction over the time horizon T will be the cumulative sum of all the received different satisfactions $S_T(x|\theta) = \sum_{t=1}^{T} \exp^{-||x_t \cdot \theta||^2}$. In our problem, we would like to maximize both the likelihood and the satisfaction. Thus, the adopted global objective function L_t (opposed to instantaneous objective function ℓ_t) to minimize will the negative log-likelihood with sum of individual log-satisfaction:

$$\begin{aligned} L_T(\theta, x, y) &= \sum_{t=1}^{T} \ln(1 + \exp -Y_{ij}^t (x_t \cdot \theta)) + \frac{\lambda}{2} ||x_t \cdot \theta||^2 \\ &= \sum_{t=1}^{T} \ell_t(\theta, x, y) \end{aligned}$$

λ can be interpreted as the importance of the satisfaction in the matching system. It captures the trade-off between exploration and exploitation in our problem. Its importance will be pointed out later when it will appear in the selection rule for J_t. In the objective function, we recognize the logistic function, plus a L_2-norm penalization. As pointed out in the first chapter, this regularization is very important to ensure existence and feasibility of the MLE. Otherwise, some assumptions such as Assumption 1 should be made. The second advantage of that L_2-regularization is that it gives some very convenient properties for our objective functions ℓ_t. These objective functions are λ-strictly convex functions and L-smooth functions. These properties will be of particular interest in the analysis of the convergence of the gradient descent scheme.

4.2 Matchmaking policy and Gradient descent

We build in the previous subsection our objective function, our target to be maximized. Its construction is based on the idea that matchmaking policies that are designed on better estimates for the different strengths of the player will result in more satisfying games. We believe that knowing the levels of players with high probability will enforce the regret to be small. The matchmaking policy we boild is made to have the biggest increase in our objective function L_T, or biggest expected increase with respect to what we already know, i.e. with respect to the canonical filtration $\mathcal{F}_t$.

At round $t > 1$, we have an estimate θ^{t-1} for the vector of strengths θ. The conditional expectation for the reward at time with respect to $\mathcal{F}_t$ and given the fact that the chosen pair of playing players is I_t and J_t reads

$$\begin{aligned}\mathbb{E}[\ell_t(\theta^{t-1})|\mathcal{F}_t \cup \{I_t, J_t\}] = &\sigma((x_t \cdot \theta^{t-1}))\ln(1+\exp^{-(x_t\cdot\theta^{t-1})}) \\ &+ \sigma(-(x_t\cdot\theta))\ln(1+\exp^{+(x_t\cdot\theta^{t-1}}) \\ &+ \frac{\lambda}{2}||x_t\cdot\theta^t||^2\end{aligned}$$

Our first criterion for our algorithm was then to pick J_t as the player maximizing this conditional expectation, that is to say we pick the player that will be satisfying enough and which tells us enough information for our logistic exploration. However, with such first criterion, the trade off between exploration and exploitation phases will always be the same, as the coefficient λ is independent to time. This will result in an constant error in the regret in the end. We want rather a dominating exploration phase in the early hours of our algorithm then the exploitation part of the criterion will have a prominent place in the choice of J_t. Heuristically, our algorithm will look like a logistic regression in the first iterations, because we need to gather information about the strengths of the players, then at some point, the logistic regression will converge θ and we will only exploit the information we have to obtain satisfying matches and games.

Therefore, to capture this reasoning and the trade-off between exploration and exploitation, we replace λ in the criterion by (β_t) which will be a decreasing sequence of coefficient. The new criterion is then : Pick a player J_t such that the expected contributions in the objective function is minimized, i.e.

$$\begin{aligned}J_t \in \arg\min_j\ &\sigma(\theta_i^{t-1}-\theta_j^{t-1})\ln(1+\exp^{-(\theta_i^{t-1}-\theta_j^{t-1})}) \\ &+ \sigma(-(\theta_i^{t-1}-\theta_j^{t-1}))\ln(1+\exp^{+(\theta_i^{t-1}-\theta_j^{t-1})}) \\ &+ \frac{\beta_t}{2}||\theta_i^{t-1}-\theta_j^{t-1}||^2\end{aligned}$$

with (η_t) an increasing sequence.

We can draw a parallel between statistical physics, this coefficient β_t and our system. At the beginning we do not know a lot of information and the system is not in order : the energy is high and so is the temperature. But when we cool extract energy from the system, that is to say we cool it, the temperature is getting down. We extract information from the different observed games and matches information, and the system will not have enough information to offer at some point. Therefore, if we consider β_t as the analogous of the inverse of the thermal energy $\beta = \frac{1}{k_B T}$ (with k_B being the Boltzmann constant), β_t should grow over time to have the exploitation term dominate the exploration term coming from logistic regression.

We chose to estimate the strengths of the players to perform a gradient step at each iteration. Gradient descent is an optimization algorithm used to minimize some function by iteratively moving in the direction of steepest descent as defined by the negative of the gradient. In machine learning, we use gradient descent to update the parameters of our model. The update will read in a general form such as

$$\theta_{t+1} \leftarrow \theta_t - \gamma\nabla\ell_t(\theta_t)$$

where γ is the constant stepsize. This method is very simple and well known: these two reasons made us choose this algorithm as a first contribution for our global matchmaking strategy. Its simplicity can be seen in its computational complexity: the gradient descent allows to save a lot of time on calculations. Moreover, the way it is done allows for a trivial parallelization, i.e. distributing the calculations across multiple processors or machines. For all these reasons, we chose this method rather than others. Our algorithm is then summarized in the following Algorithm **??**.

4.3 Controlling the regret

In this subsection, we will assume that the strengths are an i.i.d. sample of a Gaussian distribution and that kernel function used to model the satisfaction is a Gaussian kernel function. We will now try to control the regret of our algorithm. We provide a first bound on it, and how we managed to obtain such bound. Some improvements on the bound are possible and we point out in the end some aspects where one could enhance the performance and the sharpness of the bound. Our strategy will be to expand via a Taylor expansion our instantaneous regret, then control over all the terms. We will then see that the control on the regret is equivalent in fact to the control over the convergence of the logistic regression. Using results taken from the work of Francis Bach [3], we will then be able to prove a first bound on the regret.

Recall that the regret reads $R_T = \sum_{t=1}^{T} K(\theta_{I_t}, \theta_{j^*}) - K(\theta_{I_t}, \theta_{J_t}) = \sum_{t=1}^{T} r_t$. We would like to study this instantaneous regret r_t. Before doing a Taylor expansion on it to boudn the difference between the two terms, one should be careful and see that this expansion should not be done on the function $K(\theta_{I_t}, \cdot)$ between θ_{j^*} and θ_{J_t} because there is a problem of symmetry. The side of θ_{I_t} on which both θ_{j^*} and θ_{J_t} lie does not matter. Imagine the situation where $\theta_{j^*} = \theta_{I_t} + \varepsilon$ and $\theta_{J_t} = \theta_{I_t} - \varepsilon$: the regret should be 0 but the Taylor expansion will give us a non zero instantaneous regret.

To alleviate this burden, we will expand the function $K_t : x \mapsto K(|\theta_x - \theta_{I_t}|, 0)$. The Taylor expansion of K_t between j^* and J_t reads then (we omit the parameter σ in the following)

$$r_t = |K_t(\theta_{j^*}) - K_t(\theta_{J_t})| \leq |\theta_{j^*} - \theta_{I_t}| \exp^{-|\theta_{j^*} - \theta_{I_t}|^2} \Delta_t + |R(J_t)| \tag{6}$$

where $\Delta_t = |\theta_{J_t} - \theta_{I_t}| - |\theta_{j^*} - \theta_{I_t}|$ and the term $R(J_t)$ remaining in the expansion checks the following inequality due to the fact that the second derivative of K_t is continuous on the segment, thus bounded by M.

$$|R(J_t)| \leq \frac{M\Delta_t^2}{2}$$

The term Δ_t is of particular importance as it captures the spread of our strengths. In particular, in order to bound the first and second term in the sum forming the instantaneous regret, we will need to bound $\mathbb{E}[\Delta_t^2]$. Moreover, we will need to take care of the term $|\theta_{j^*} - \theta_{I_t}| \exp^{-|\theta_{j^*} - \theta_{I_t}|^2}$ in expectation. To do so, we will use the Cauchy Schwartz and prior to do need to bound $\mathbb{E}[|\theta_{j^*} - \theta_{I_t}| \exp^{-|\theta_{j^*} - \theta_{I_t}|^2}]$ by a study of Gaussian order statistics.

We remark that

$$|\theta_{j^*} - \theta_{I_t}| = \min(\theta_{(k+1)} - \theta_{(k)}, \theta_{(k)} - \theta_{(k-1)})$$

for a special $k \in [n]$. However, the Gaussian order statistics are not easy to handle. Similarly to what Boucheron and Thomas did in [5], we use a Rény's transformation. After calculations we have

$$\begin{aligned} |\theta_{j^*} - \theta_{I_t}| \leq \big(U(\theta_{(k+1)}) - U(\theta_{(k)})\big) \sup_{x \in [U(\theta_{(k)}), U(\theta_{(k+1)})]} U^{-1}(x) \\ + \big(U(\theta_{(k)}) - U(\theta_{(k)})\big) \sup_{x \in [U(\theta_{(k)}), U(\theta_{(k+1)})]} U^{-1}(x) \end{aligned}$$

The random variables of the exponential spacings $U(\theta_{(k+1)}) - U(\theta_{(k)}$ are all independent and following exponential distribution. Moreover, both supremums are actually maximums as the function is continuous over a compact, and these maximum are supposed to be bounded by N. Thus, the expectation $\mathbb{E}[|\theta_{j^*} - \theta_{I_t}| \exp^{-|\theta_{j^*} - \theta_{I_t}|^2}]$ is bounded such that

$$\mathbb{E}[|\theta_{j^*} - \theta_{I_t}| \exp^{-|\theta_{j^*} - \theta_{I_t}|^2}] \leq \mathbb{E}[|\theta_{j^*} - \theta_{I_t}| \leq 2N \tag{7}$$

The next step would be to control the term $\Delta_t = |\theta_{J_t} - \theta_{I_t}| - |\theta_{j^*} - \theta_{I_t}|$. Our goal is to prove that the control over Δ_t is equivalent to the control over the convergence of the logistic regression. Using the triangle inequality and the criterion, we end up with the following bound for the quantity Δ_t

$$\boxed{|\Delta_t| \leq |\theta_{J_t} - \theta^t_{J_t}| + |\theta^t_{j^*} - \theta_{j^*}| + \frac{1}{1 - \frac{K}{\beta_t}} \big[|\theta^t_{j^*} - \theta_{j^*}| + \underbrace{|\theta_{j^*} - \theta_{I_t}|}_{\sim Exp(k)} + |\theta^t_{I_t} - \theta_{I_t}|\big]}$$

As explained before, we will now use a result derived by Francis Bach in [3] to conclude our bound on the regret. As explained in this work, the control over the convergence of the logistic regression on the stochastic regime is based on the particular properties of convexity of the logistic loss. Let ℓ be the full logistic function $\ell : x \mapsto \ln(1 + e^{-x})$, the instantaneous logistic functions form a stochastic sequence of function $\ell_t : \theta \mapsto \ln(1 + e^{-y_t(x_t \cdot \theta)})$. The main assumptions that the objective functions should fulfill are

1. ℓ and ℓ_t should be differentiable and convex.
2. ℓ and ℓ_t should be L-smooth, and gradients are bounded by R.
3. ℓ_t should be $\mathcal{F}_t$ measurable.
4. for all θ_1, θ_2, the function $\Phi : t \mapsto \ell(\theta_1 + t(\theta_2 - \theta_1))$ satisfies: $\forall t \in \mathbb{R}, |\Phi'''(t)| \leq L||\theta_1 - \theta_2||\Phi''(t)$ (Self-concordance [2])

The self concordance property was developed by Bach in [2]. This notion of self concordance is an important tool in convex optimization and in particular for the study of Newton's method [16]. The key consequence of our notion of self-concordance is that we can replace global strong-convexity constant by the local strong convexity constant, the global one is indeed zero here. The logistic function is indeed fulfilling these assumptions. We have a constant step-size but the estimate $\hat{\theta}$ we will use to proxy θ is an average of the different θ^t compared to previously, i.e.

$$\hat{\theta}^T = \sum_{s=1}^{T} \theta^t$$

Theorem 4. Assume $\gamma = \frac{1}{2R^2\sqrt{T}}$. Let $\mu > 0$ be the lowest eigenvalue of the Hessian of ℓ at the unique global optimum θ^*. Then it exists $C > 0$, depending on the initialization, such that

$$\mathbb{E}[||\theta^T - \theta||] \leq \frac{R}{\sqrt{T}\mu} C$$

With this bound on the convergence of the logistic regression, we obtain the following bounds for Δ_t and Δ_t^2 computed with $\hat{\theta}^t$ and not θ^t (using $\mathbb{E}[X^2] \leq \mathbb{E}[X]^2$ are in expectation:

$$\mathbb{E}[\Delta_t] \leq (3 + \frac{1}{1 - K/\beta_t}) \frac{R}{\sqrt{T}\mu} C \tag{8}$$

$$\mathbb{E}[\Delta_t^2] \leq (3 + \frac{1}{1 - K/\beta_t})^2 \frac{R^2}{T\mu^2} C^2 \tag{9}$$

Hence, we can now bound the first term of the instantaneous regret r_t via the Cauchy-Schwarz inequality

$$\mathbb{E}[|\theta_{j^*} - \theta_{I_t}| \exp^{-|\theta_{j^*} - \theta_{I_t}|^2} \Delta_t] \leq 2N(3 + \frac{1}{1 - K/\beta_t})^2 \frac{R^2}{t\mu^2} C^2$$

Finally we derive the following bound on the regret for our algorithm

$$\boxed{\mathbb{E}[R_T] \leq (M/2 + 2N)(3 + \frac{1}{1 - K/\beta})^2 \frac{R^2 C^2}{\mu^2} \ln T}$$

4.4 Further directions

We have now a first bound on our regret that is scaling very well in $O(\ln T)$ which is very sharp for time dependence. Even though this results seems a bit optimistic, it is a first step in the analysis of this algorithm. The dependence in time is captured but not so is the dependence in the number of players which is of particular interest in our study. The next step of this line of work would be for instance to understand how the different constants scale with n such as

- M the upper bound of the second derivative of the logistic loss in the Taylor expansion.
- N the upper bound in the Rény's transformation made in equation 7. It is likely that this bound has a dependency in n as it should bound n different supremums.
- C the bound coming form the work of Bach [3]. In this constant is hidden the initialization and the term $||\theta_0 - \theta^*||$.
- K the upper bound of the Lipschitz constant.

Moreover, the thermodynamic dynamics were not leverage in the derivation concerning the regret. It is a bit weird as this seemed very important in our heuristic reasoning. A next direction would be to understand deeply the impact and influence of this sequence of temperature (β_t). We can expect a dependency in time such that $\beta_t \propto \frac{1}{t}$ or $\beta_t \propto \frac{1}{\sqrt{t}}$, this should play a prominent role in the choice of player J_t depending on the time we are at.

5 Conclusion

Coming from the world of Video games, our matchmaking problem was a source of challenges that we tried to face or at least to overcome. The setting seemed quite simple at first sight: a queue of players is waiting to play with each other and we want to make the system able to match satisfyingly people.

Algorithm	Time	n	Computations
Exp-3	$\tilde{O}(\sqrt{T})$	$\tilde{O}(\sqrt{n^2 \ln n})$	feasible
ε-Greedy	$O(T^{2/3})$	$\tilde{O}(\sqrt{n \ln n})$	feasible
PolicyElimination	$\tilde{O}(\sqrt{T})$	$\tilde{O}(\sqrt{n^2 \ln n})$	impossible
Ilovetoconbandits	$\tilde{O}(\sqrt{T})$	$\tilde{O}(\sqrt{n^2 \ln n})$	feasible
Log-reg	$\tilde{O}(\ln T)$	??	feasible

Table 1: Recap of the different performance for the studied algorithms

The first step was to formalize the problem in a rigorous manner. We decide to adopt an online setting in the class of Bradley -Terry models for the outcome of the different games. We chose to model satisfaction by a kernel function for their symmetric positive properties and our performance criterion was regret as we wanted our solution to be maximizing the online satisfaction compared to the offline satisfaction.

Then, we focused on the particular problem of contextual bandits that is well suited for this problem of matchmaking. We present different classical algorithms that will make our baseline.The latter will be our target to be outperformed by our own technique. The ε-greedy algorithm appears to be the most efficient algorithm in terms of time-dependence and dependence in the number of players. We present the results of the different dependency in the following table 1. Even though, exp-3 seems to have a better statistical dependency as the ε-greedy algorithm, it does not scale well with respect to the number of players.

Beyond those classical methods, we decide to tailor the exploration phase in our algorithm. We then studied two different algorithms PolicyElimination and Ilovetoconbandits, that have both great statistical performance. However only the second one can be implemented in practice. Those two algorithms were added to the list of the benchmark algorithms that should be broken by our algorithm.

Finally, the third time of this master's thesis was to design and develop a matchmaking algorithm capable to break those baseline. We imagined an algorithm based on the idea that having sharp information about the strengths of the players would make easier to decide with whom an incoming partner should play. Thus, the designed algorithm is a gradient descent algorithm with a loss function combining the logistic function with an L_2-regularization capturing the satisfaction. Therefore, the control on the pseudo-regret is very linked to the convergence of an online logistic regression. We derived a bound on this regret, however a lot of information is hidden in constants. A next step would be to understand the different constants involved in this bound.

We hope that improvements in the analysis and understanding of our own designed algorithm will be found and that this line of work would be the beginning of an efficient method that will be used in different video game systems.

References

[1] Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert Schapire. Taming the monster: A fast and simple algorithm for contextual bandits. volume 32 of *Proceedings of Machine Learning Research*, pages 1638–1646, Bejing, China, 22–24 Jun 2014. PMLR.

[2] Francis Bach. Self-concordant analysis for logistic regression. *Electron. J. Statist.*, 4:384–414, 2010.

[3] Francis Bach. Adaptivity of averaged stochastic gradient descent to local strong convexity for logistic regression. *J. Mach. Learn. Res.*, 15(1):595–627, January 2014.

[4] Alina Beygelzimer, John Langford, Lihong Li, Lev Reyzin, and Robert Schapire. Contextual bandit algorithms with supervised learning guarantees. volume 15 of *Proceedings of Machine Learning Research*, pages 19–26, Fort Lauderdale, FL, USA, 11–13 Apr 2011. JMLR Workshop and Conference Proceedings.

[5] Stéphane Boucheron and Maud Thomas. Concentration inequalities for order statistics. *Electronic Communications in Probability*, 17(0), 2012.

[6] Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

[7] Sébastien Bubeck. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. 2012.

[8] Miroslav Dudik, Daniel Hsu, Satyen Kale, Nikos Karampatziakis, John Langford, Lev Reyzin, and Tong Zhang. Efficient optimal learning for contextual bandits, 2011.

[9] L. R. Ford. Solution of a ranking problem from binary comparisons. *The American Mathematical Monthly*, 64(8):28–33, 1957.

[10] Geoffrey J. Gordon. Regret bounds for prediction problems. In *Proceedings of the Twelfth Annual Conference on Computational Learning Theory*, COLT '99, page 29–40, New York, NY, USA, 1999. Association for Computing Machinery.

[11] J. Hannan. Approximation to bayes risk in repeated play. *Contributions to the theory of games*, 1957.

[12] Franz J. Király and Zhaozhi Qian. Modelling competitive sports: Bradley-terry-Élő models for supervised and on-line learning of paired competition outcomes, 2017.

[13] Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. *Proceedings of the 19th international conference on World wide web - WWW '10*, 2010.

[14] J.I. Marden. *Analyzing and Modeling Rank Data*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis, 1996.

[15] Issei Matsumoto, Kohei Hatano, and Eiji Takimoto. Online density estimation of bradley-terry models. In Peter Grünwald, Elad Hazan, and Satyen Kale, editors, *Proceedings of The 28th Conference on Learning Theory*, volume 40 of *Proceedings of Machine Learning Research*, pages 1343–1359, Paris, France, 03–06 Jul 2015. PMLR.

[16] Y. Nesterov and A. Nemirovskii. *Interior-point Polynomial Algorithms in Convex Programming*. Studies in Applied Mathematics. Society for Industrial and Applied Mathematics, 1994.

[17] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, page 928–935. AAAI Press, 2003.