Gorthaur-EXP3: Bandit-based selection from a portfolio of recommendation algorithms balancing the accuracy-diversity dilemma

Nicolas Gutowski, Tassadit Amghar, Olivier Camp, Fabien Chhel

Gorthaur-EXP3: Bandit-based Selection from a Portfolio of Recommendation Algorithms Balancing the Accuracy-Diversity Dilemma

Nicolas Gutowski[a,b], Tassadit Amghar[b], Olivier Camp[a], Fabien Chhel[a]

[a] *ESEO-TECH / ERIS*
10 boulevard Jean Jeanneteau
49000 Angers, France
[b] *LERIA / Université d'Angers*
2 boulevard de Lavoisier
49000 Angers, France

Abstract

Nowadays, real-world pervasive computing applications increasingly face multi-objective problems. This is the case for recommendation systems where, from a user's view point, recommended items must be both accurate and diverse.

In recent years, model-based recommendation systems like those relying on *Multi-Armed Bandit* algorithms have been extensively studied. They are known to ensure theoretical guarantees of global accuracy. Nevertheless, despite these guarantees, the existing algorithms obtain different results depending on the application or on the dataset they operate on. Hence, when one needs to integrate such solutions, they should first be thoroughly evaluated to ensure the chosen method is efficient for the dynamic and potentially non-stationary nature of the target environments. However, human-based evaluations cost in time and money. Here, we propose a novel algorithm portfolio approach, *Gorthaur-EXP3* aiming at automatically selecting the optimal algorithms which best maximise global accuracy and diversity of recommendations according to a predefined trade-off. Our method uses the *EXP3* bandit algorithm which ensures a continuous exploration and a systematic exploitation of the best algorithm to apply in each situation it encounters. *Gorthaur-EXP3* is an extension of the original *Gorthaur* method, which uses a roulette wheel selection, and obtains better results in most experimental cases.

Keywords: Application of Reinforcement Learning; Recommendation Systems;

Multi-Armed Bandit; Contextual Multi-Armed Bandit; Portfolio Approach
2020 MSC: 68T01, 68T05, 68T20, 90C29

1. Introduction

In recent years, bandit-based recommendation systems have been widely evaluated offline and online [1, 2, 3, 4, 5]. Whether they are contextual (*Contextual Multi-Armed Bandit : CMAB*) [6] or not (*Multi-Armed Bandit : MAB*) [7], they rely on theoretically grounded proofs and offer strong guarantees for various applications.

Nevertheless, *MAB* and *CMAB* algorithms [1, 8, 9] give different experimental results [10] for both global accuracy and diversity [11] depending on the dataset they use. Those differences can be explained by the nature of the data on which the algorithms are trained, for instance their fairness [12] (e.g., calibration within groups, balance for the negative class, balance for the positive class) or the presence of intrinsic biases. Indeed, since recently, the 2018 Turing Award winner Yann Lecun argues that *"Data is biased, in part because people are biased. But learning algorithms themselves are not biased. Bias in data can be fixed. Bias in people is harder to fix."* Such bias in datasets can partly explain the variety of results that are obtained by identical algorithms on different data.

The above considerations that are true for both offline and online applications, led us to develop a novel approach for the bi-objective meta-selection of learning algorithms from a portfolio. The first version of our approach entitled *Gorthaur* and presented in [13], aimed at maximising both criteria of global accuracy and diversity of the recommendations made to users. The principle of *Gorthaur* was to use a roulette wheel to select algorithms proportionally to their ability to maximise the two criteria according to a desired predefined trade-off. We will now refer to the original *Gorthaur* as *Gorthaur-Wheel*. This heuristic-based meta-selection approach chooses, from a given set or portfolio of algorithms, the ones which are the most adapted to reach the desired trade-off (accuracy/diversity) for a given situation. *Gorthaur-Wheel* obtains consistent results which have been compared and validated with those of each algorithm in the portfolio. However, even though *Gorthaur-Wheel* obtains encouraging results, it still suffers from an important loss of global accuracy compared to the best algorithm in the portfolio. Depending on the requirements of the recommendation system in which the method is to be integrated, it could be wise to explore another algorithm selection strategy. We believe that the theoretical guarantees provided by a model-based selection mechanism, should enhance the method and provide more robustness.

Hence, in this article we propose a novel algorithm selection approach for *Gorthaur-EXP3* and compare it with the original *Gorthaur-Wheel* method [13] in terms of global accuracy and diversity. *Gorthaur-EXP3* is similar to *Gorthaur-Wheel* but uses an *EXP3* bandit algorithm to select algorithms among its portfolio. Thus, *Gorthaur-EXP3* ensures that the optimal algorithm that best meets the optimization criteria ends up being systematically chosen.

Herein, we focus on the global accuracy and diversity results obtained by *Gorthaur-EXP3*'s algorithm selection approach in different use cases. They are evaluated and compared to those obtained by *Gorthaur-Wheel* in different cases: stationary environment and non stationary environment; using a mechanism that shares rewards among the algorithms of the portfolio and with no reward sharing mechanism.

We observe that, for most experiments, *Gorthaur-EXP3* obtains better results than the original *Gorthaur-Wheel* method:

1. In stationary environments, whether rewards are shared or not, *Gorthaur-EXP3* systematically outperforms the original *Gorthaur-Wheel* method except for the non contextual case.
2. In non stationary environments, when rewards are not shared, *Gorthaur-EXP3* systematically outperforms the original *Gorthaur-Wheel* method.
3. In non stationary environments, when rewards are shared, *Gorthaur-EXP3* is better than the original *Gorthaur-Wheel* method for concept-drift/shift cases. This is not the case for covariate-shift for which the original *Gorthaur-Wheel* obtains better results.

The paper is organised as follows. Section 2 clarifies our motivations along with the problem. Section 3 reviews the related work on *MAB* and *CMAB* problems for recommendation and sheds light on meta-selection of learning algorithms. This section also presents preliminary studies that were carried out to evaluate the individual algorithms included in the portfolio of recommendation algorithms. Section 4 describes the method we propose: *Gorthaur-EXP3*. Section 5 depicts our experimental evaluation; the results of which are presented and discussed in Section 6. Finally, Section 7 presents our conclusion and perspectives.

2. Motivation and problem

The experiments that have been carried out to evaluate *MAB* or *CMAB* based recommendation systems have shown that no single algorithm is suited for all situations. Of course, *CMAB* algorithms are those that fit best when contextual information is available, while *MAB* algorithms should be preferred when it is missing. However, performances also depend on cases that are encountered by recommendation systems. For instance, some methods perform better than others when facing non-stationary environments where the rewarding distribution changes over time.

Moreover, in the case of recommendation systems, we believe that the accuracy metric which reflects the proportion of successful arm selections (or recommendations) is not sufficient to evaluate the performance of the system and should be extended with a diversity metric. Indeed, by diversifying the recommendations it provides, the system is more likely to explore new actions and is more able to discover new relevant items that fit user preferences, through serendipity.

Thus, we advocate that, a multi-armed bandit based recommendation system should not solely rely on one *MAB*/*CMAB* algorithm for all its recommendations, but rather should choose an algorithm from a pool of available methods for each recommendation it provides depending on the data and situation it is faced with. In this work we propose to provide the recommendation system with a portfolio of *MAB* and *CMAB* algorithms from which to choose for making its recommendations. We propose to consider the algorithm selection as a bi-objective optimization problem in which a target trade off between global accuracy and diversity of recommendations should be reached. We have carried out preliminary studies of such a portfolio approach using a roulette wheel selection mechanism based on the fitness of each algorithm in the portfolio to meet the accuracy/diversity trade-off. We obtained very interesting first results [13]. In this article, we propose to enhance the method by considering its algorithm selection phase as a *MAB* problem and using a dedicated algorithm to solve it.

3. Related work and background

In this section we start by reminding the non-contextual and contextual Multi-Armed Bandit (*MAB* and *CMAB*) problems, and the main algorithms used to solve them. In a second part we present how recommendation can be seen as a Multi-Armed Bandit problem. With this approach, items to be recommended represent the arms of the *MAB* or *CMAB* problem and the aim of the problem is to select an item that is most likely to best fit the recommendation request, using an appropriate algorithm.

3.1. Multi-Armed Bandits and Recommendation

The Multi-Armed Bandit (*MAB*) problem has been extensively studied since its first formulation by [7] in 1952.

The problem can be illustrated simply by considering a player facing slot machines (one-armed bandits) in a casino. The player's aim is to pull the most rewarding arm each time a coin is inserted in the slot machine. The *MAB* problem refers to the challenge of developing a strategy aimed at determining, at each turn, an arm to pull (without any initial prior knowledge of the payoff rate of each of them) in order to maximise the total gain. One must thus find a trade-off between the exploration needed to estimate the value of each arm, and the exploitation which consists in relying on the knowledge learnt from past experiences to select the best rewarding arms.

An extended version of this problem, known as the Contextual Multi-Armed Bandit Problem *CMAB* [6, 9], takes contextual information into account. More precisely, with *CMAB* problems, it is considered that the reward of an arm depends on the context. The challenge thus becomes, at each turn, to choose the best rewarding arm according to the context.

Several solutions to the *MAB* and *CMAB* problems have been developed: some using stochastic formulations [14, 15], others using Bayesian [16] approaches. *MAB* and *CMAB* problems have been adapted to fit the requirements of recommendation systems. Thus, several works have used *MAB*s to model recommendation problems and experimented the use of algorithms such as *UCB* [14], *Thompson Sampling* (*TS*) [16] and *EXP3* [8] to solve them. Similarly, contextual recommendation problems, in which the fitness of a recommendation depends on the context, have been modeled as *CMAB*s.

Algorithms such as *LinUCB* [1], *Linear Thompson Sampling* (*LinTS*) [9] have been widely experimented to solve them.

The following subsection formally describes the *MAB* and *CMAB* problems for recommendation and depicts the notion of regret in such settings.

3.1.1. Multi-Armed Bandits for recommendation

Hereafter, based on the original definition of the Multi-Armed Bandit (*MAB*) problem, we formally describe the *MAB* problem for recommendation, algorithms that solve this problem and the notion of regret.

The Multi-Armed Bandit (*MAB*) problem for recommendation. Let $A = \{a_1, \ldots, a_k\}$ be a set of k independent arms to be pulled. In the specific case of recommendation, there exists an item to be recommended corresponding to each arm. Let D_r denote the distribution of the reward expectancy of the items to be recommended. Therefore, $D_r = (\mu_{a_1}, \ldots, \mu_{a_k}) \in [0,1]^k$, where $\mu_{a_i} \in [0,1]$ is the reward probability when recommending item a_i, $i \in [1,k] \cap \mathbb{N}$. The problem is sequential: at each iteration $t \in [1,T]$, a user $u \in U$ arrives and is considered. First, a sample $(r_{a_1}, \ldots, r_{a_k}), r_{a_i} \in \{0,1\}$ is drawn from D_r. Then one item $a_i \in A$ is chosen by the player (the recommender) and recommended to u. Finally, user u's reward r_{a_i} is revealed: 1 if the user appreciated the recommended item and 0 otherwise.

MAB ***algorithms for recommendation.*** At each iteration t, a *MAB* algorithm $\mathcal{A}$ for recommendation determines an item $a_i \in A$, $A = \{a_1, \ldots, a_k\}$ to recommend, based on the previous sequence of $t-1$ observations $(a_{i,1}, r_{a_i,1}), \ldots, (a_{i,t-1}, r_{a_i,t-1})$. On receiving the recommendation the user evaluates it and returns a feedback to the player (the recommender) was a success and 0 otherwise. Upon receiving the user's feedback the player updates the rewards vector $\vec{r}_t$.

The player's goal is to maximise the expected total reward $\sum_{t=1}^{T} \mathbb{E}_{\vec{r}_t \sim D[r_{a,t}]}$. An optimal policy knows each item's average reward and recommends item a^* with the highest average reward, i.e., $a^* = \arg\max_{a \in A}(\mu_a)$.

Thus, in order to determine the efficiency of a *MAB* algorithm $\mathcal{A}$ for recommendation, we should measure the cumulative regret it obtains $\rho_T(\mathcal{A})$ (where T is the Horizon) and compare it to that obtained by the optimal policy. We can therefore define the regret as follows.

Regret in* MAB *problems for recommendation. In the case of bandit-based recommendation systems, a gain can be considered as a successful recommendation to a user u, and a regret as a failure. Let $g_T^* = \sum_{t=1}^{T} r_{a^*,t}$ the gain obtained by an optimal policy at horizon T. Then, the gain expectancy of the optimal policy is $\mathbb{E}[g_T^*] = T\ \mu^*$. Let $\mathcal{A}$ be a *MAB* algorithm for recommendation. The cumulative regret of an algorithm having recommended the following sequence of items $a_{i,1}, \ldots, a_{i,T}$ is therefore $\rho_T = g_T^* - \sum_{t=1}^{T} r_{a_i,t}$.

3.1.2. Contextual Multi-Armed Bandits for recommendation

Herein, from the original Contextual Multi-Armed Bandit (*CMAB*) problem definition proposed by Langford in 2008 [6], we formally describe the *CMAB* problem for recommendation, algorithms that solve this problem, and the notion of regret when using a context-aware approach.

Contextual Multi-Armed Bandit (*CMAB*) problems for recommendation. Let $A = \{a_1, \ldots, a_k\}$ be a set of k independent arms to be pulled, for each of which, in the case of recommendation, there exists an item to be recommended. Let $D_{x,r}$ denote the joint distribution between contexts x and rewards r, such that $D_{x,r} = (x, r_{a_1}, \ldots, r_{a_k})$, where $x \in X \cap \mathbb{R}^d$ is a context, and $r_{a_i} \in \{0, 1\}$ is the reward associated to item a_i, $i \in [1, k] \cap \mathbb{N}$. The problem is sequential: at each iteration t, a sample $(x, r_{a_1}, \ldots, r_{a_k})$ is drawn from $D_{x,r}$, user u with his/her context x arrive and are observed, then an item a is selected by the player, recommended to u and its reward r_a is revealed (1 if the user appreciated the recommended item, 0 otherwise).

CMAB *algorithms for recommendation.* At each iteration t a *CMAB* algorithm $\mathcal{A}$ determines an item to recommend $a_i \in A$, $A = \{a_1, \ldots, a_k\}$. This choice is based on the previous sequence of observations $(x_1, a_{i,1}, r_{a_i,1}), \ldots, (x_{t-1}, a_{i,t-1}, r_{a_i,t-1})$ and the current observed context x_t. The algorithm then updates the rewards vector $\vec{r}_t$ according to the user's feedback.

The goal is to maximise the expected total reward $\sum_{t=1}^{T} \mathbb{E}_{x,\vec{r}_t \sim D[r_{a,t}]}$. Let $\Pi : X \to A$ be the set of possible recommendation policies where the optimal policy to be determined is $\pi^* = \arg\max_{\pi\in\Pi} \mathbb{E}_{r,x}[r_{t,\pi(x)}]$. Thus, in order to determine the efficiency of a *CMAB* algorithm $\mathcal{A}$ for recommendation, we can measure the cumulative regret it obtains $\rho_T(\mathcal{A})$ and compare it to that obtained by the optimal recommendation policy. We can therefore define the regret as follows.

Regret in* CMAB *problems for recommendation. In the case of context-aware bandit-based recommendation systems, a successful recommendation given to a user u in context x is considered a gain; whereas, a failed recommendation is a regret. The expected reward for a recommendation policy $\pi \in \Pi$ is:

$$R(\pi) = \mathbb{E}_{(x,\vec{r})\sim D}[r_{\pi(x)}]$$

Consider any *CMAB* algorithm $\mathcal{A}$. Let $Z^T = \{(x_1, \vec{r_1}), \ldots, (x_T, \vec{r_T})\}$, and the expected regret of $\mathcal{A}$ with respect to policy π be:

$$\Delta\rho(\mathcal{A}, \pi, T) = T\ R(\pi) - \mathbb{E}_{Z^T\sim D^T} \sum_{t=1}^{T} r_{\mathcal{A}(x),t}$$

The expected regret of $\mathcal{A}$ up to horizon T with respect to the recommendation policy space Π is then defined as:

$$\Delta\rho(\mathcal{A}, \Pi, T) = \sup_{\pi\in\Pi} \rho(\mathcal{A}, \pi, T)$$

3.1.3. *Limits of* MAB *and* CMAB

Despite the theoretically grounded guarantees that *MAB* and *CMAB* algorithms ensure, we observe different results depending on the nature of the real-world applications or the offline datasets [13]. Thus, when one needs to deploy a machine learning model-based recommendation system in the real-world, it is necessary to consider beforehand which algorithm could best meet the application needs. Moreover, we should ensure the system provides both accurate and diversified recommendations.

3.2. *Meta-selection of learning algorithms*

In order to maximize the performance of online recommendation systems, a solution could consist in choosing the best algorithm from the state-of-the art. Unfortunately, without prior knowledge, there is no "*best*" algorithm for all cases. We advocate that a selection mechanism should be used to dynamically choose the algorithm that best fits the application or dataset.

The problem of selecting an efficient algorithm was introduced by [17] as a general scheme. Given a problem space and an algorithm space, the basic idea is to find an algorithm/problem pair that identifies the best algorithm for the specific problem (instance). In the case of combinatorial optimization, algorithm selection has been extensively studied over the past two decades [18] and has obtained competitive results for the SAT problem [19].

Reinforcement learning is one approach used by meta-heuristic algorithms for parameter tuning. For example in [20], *UCB*-based algorithms are used to provide better strategies for the Adaptive Operator Selection (*AOS*) issue in Evolutionary Algorithms. More precisely, the authors present several parameter control mechanisms to tackle the selection of mutation and crossover operators during the online-phase. Each operator is considered as an arm and the reward function (credit assignment) is computed. To avoid confusion, we consider the diversity in the prism of recommendation systems, i.e., where it corresponds to the diversity of the arms chosen by the *MAB/CMAB* algorithms and not to the diversity of the population (e.g., entropy between individuals).

In the case of *MAB* problems and considering an online evaluation, several approaches have been developed and meta-selection of learning algorithms can be summarized to the dynamic selection of an algorithm in a portfolio [21, 22, 23]. To implement this approach, first, the set of algorithms contained in the portfolio needs to be chosen. This choice is presented and discussed in section 3.5. Then, an adequate algorithm selection mechanism should be defined. We now present two selection strategies which inspired our approach.

3.2.1. Selection of learning experts

In [24], the authors present a novel meta-selection approach based on the selection of what they refer to as "learning experts" using *EXP3*. They consider a learning expert as being a contextual multi-armed bandit algorithm. At each iteration, a player chooses the expert that selects an action after observing the context vector representing the environment. Each expert aims at minimizing its own estimation error and the player estimates the experts' performances.

The authors argue that exploration is done by an efficient grounded algorithm and the bias of experts, e.g., sensibility to the context, is reduced by selecting the best one. Two algorithms are proposed to handle meta-selection: *Learn, Then Explore and Exploit - LTEE* and *Learn, Explore and Exploit - LEE*. *LTEE* is a 2-steps algorithm. During the first step, the learning phase, each expert can be selected and evaluated. For the second step, an optimal expert is elected after having successively eliminated experts according to identified rejection criteria. Despite good performances in theory, in practical cases the complexity can be impacted negatively by the complexity of the worst expert. To improve the results, with the *LEE* algorithm, learning, exploration and exploitation are handled in parallel and done simultaneously.

3.2.2. Multi-objective selection with Gorthaur-Wheel

To deal with multiple performance criteria, Multi-Objective Multi-Armed Bandit methods [25] such as *MO-MAB* [26, 27] or *MOC-MAB* [28] (contextual version) have been proposed. With classical *CMAB* problems the agent aims at maximizing its cumulative reward on a single-objective. Contrarily, with *MOC-MAB* the agent aims at maximizing its cumulative reward for a non-dominant objective while ensuring that it also maximizes the cumulative reward it obtains for a dominant objective. However, those approaches aggregate a set of performance criteria into one. Hence, inducing a mono-objective resolution.

Moreover, the dominant and non-dominant objectives may be conflicting and thus, the maximization of both objectives can be detrimental for the performance of an entire offline simulation or a whole online evaluation. Hence, to ensure keeping the performance of each of the selected *MAB* or *CMAB* algorithms without having to deal with contradictory objectives, we preferred to implement a portfolio method [13] which selects the algorithms that best fit the bi-objective trade-off that is set *a priori* according to application requirements.

In [13], we have implemented a bi-objective portfolio approach aimed at selecting bandit-based recommendation algorithms by maximising two criteria: Global accuracy and diversity. This method entitled *Gorthaur-Wheel*, uses a roulette wheel to dynamically select multi-armed bandit algorithms used for recommendation. According to the first results obtained, we observe that the advantage of using *Gorthaur-Wheel* is twofold:

1) The method manages to find a trade-off in cases where there is no prior knowledge about the nature of the dataset or the recommendation application to deploy; 2) For a given dataset, *Gorthaur-Wheel* is able to identify a set of optimal algorithms.

Nevertheless, depending on the needs of the recommendation system, it may be preferable to use a Multi-Armed Bandit selection instead of a Roulette wheel. It would be the case when one prefers to select the optimal algorithm rather than a set of them proportionally. Thus, in this article we propose a novel approach *Gorthaur-EXP3* built upon the combination of the original *Gorthaur-Wheel* algorithm [13] and the *EXP3* selection of learning experts [24] (where algorithms of the portfolio are considered as experts).

Herein, our main motivations for comparing a bandit-based and a roulette wheel selection strategy are twofold:

1. We need to shed light on which method allows to best meet the two criteria that we wish to optimise in stationary environments ; i.e., global accuracy and diversity of recommendations.
2. Since online applications of recommendation systems have to deal with non-stationarity, we need to know which strategy is able to best cope with this issue.

3.3. *The original* Gorthaur-Wheel *method*

According to the literature on recommendation systems, the evaluation of multi-armed bandit algorithms are most often based on the cumulative rewards (or regret) they obtain. This means such recommendation systems aim at maximising the global accuracy, which in this specific application case can be at the expense of low diversity. The principle of *Gorthaur-Wheel* (**G**eneric-**OR**ienT**ed** **H**euristic **A**lgorithm for **U**ser **R**ecommendation) [13] relies on the use of a portfolio of *MAB*/*CMAB* algorithms for recommendation and uses a roulette wheel strategy to select them. The objective of *Gorthaur-Wheel* is to maximise the trade-off between global accuracy and diversity according to a given target. The main goal of *Gorthaur-Wheel*, inspired by the *Compass* method [29], is thus to benefit from the advantage of each algorithm in the portfolio on both these criteria.

As presented in [13] and illustrated on Figure 1, let ΔAcc denote the variation of global accuracy on the vertical axis, and ΔDiv denote the variation of diversity on the horizontal axis. At the starting point t_0, we express a reference vector $\vec{c}$ defined by the angle $\Theta \in [0; \frac{\pi}{2}]$ made by $\vec{c}$ with the horizontal axis (ΔDiv). This reference vector $\vec{c}$ expresses the trade-off required between global accuracy and diversity. According to the value of Θ we set, we can choose to favor global accuracy, diversity, or to compute a balance between both criteria.

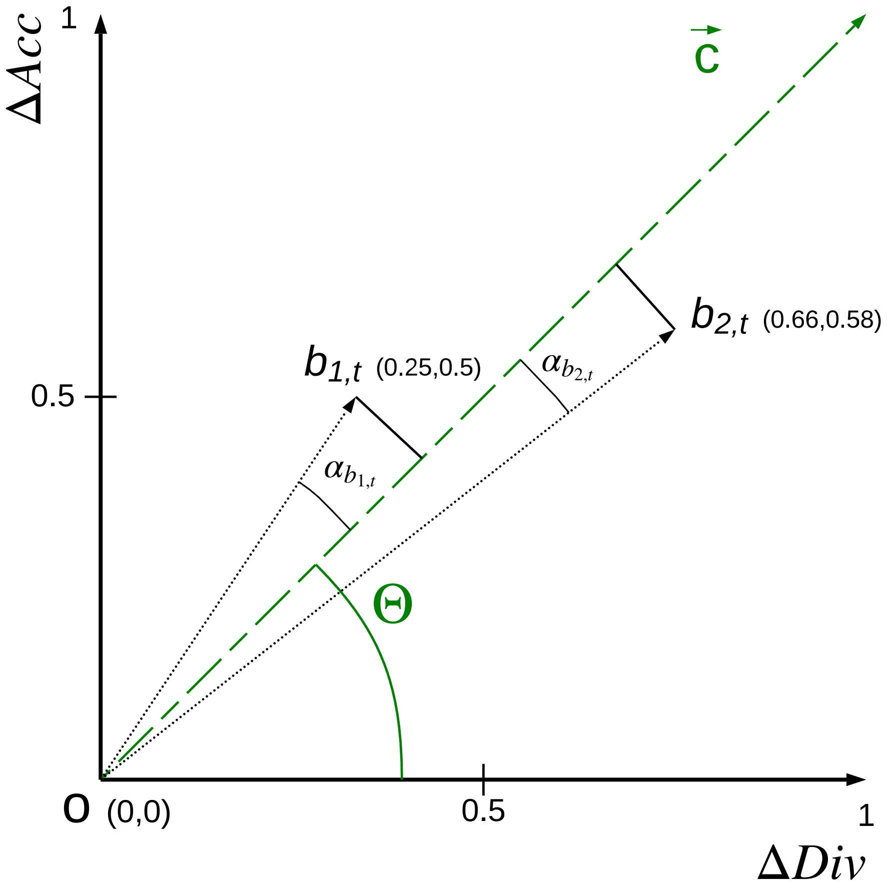


Figure 1: Gorthaur's projections on reference vector c to compute fitness

Let $\mathcal{B}$ denote a set of recommendation algorithms all having the same fixed number of arms. At each iteration t, *Gorthaur* selects an algorithm $b \in \mathcal{B}$ which itself chooses an arm a to recommend according to its own strategy. Then, *Gorthaur* computes the global accuracy $acc(b, t)$ of algorithm b at iteration t and its diversity $div(b, t)$.

$acc(b,t)$, the average global accuracy is defined by:

$$acc(b,t) = \frac{g(b,t)}{t_b} \tag{1}$$

where $g(b,t) = \sum_{t=1}^{t} r(b,t)$ is the sum of rewards obtained by algorithm b at iteration t, $\forall t \in [1;T]$, and t_b is the number of times algorithm b was selected by *Gorthaur-Wheel* since the initial time step t_0.

$div(b,t)$ is the diversity obtained by algorithm b at iteration t and is defined as follows. By considering $N_{b,t} = \{n_{a_1}(b,t), \ldots, n_{a_k}(b,t)\}$ where $n_{a_j}(b,t)$ is the total number of times arm a_j was recommended by algorithm b at time step t, we formally define $c_v(N_{b,t}) = \frac{\sigma(N_{b,t})}{\overline{N_{b,t}}}$ where $\overline{N_{b,t}}$ is the average number of times any arm has been selected by algorithm b at trial t, and $\sigma(N_{b,t})$ is its standard deviation. Then, the resulting diversity for each algorithm b at trial t is:

$$div(b,t) = 1 - \frac{c_v(N_{b,t})}{\sqrt{k}} \tag{2}$$

The ability of an algorithm to provide recommendations that are both accurate and diversified is then expressed by a vector $o_{b,t}$ such that:

$$o_{b,t} = (div(b,t), acc(b,t))$$

The measured values of $div(b,t)$ and $acc(b,t)$ are then normalised as follows:

$$div^{norm}(b,t) = \frac{div(b,t)}{max\{div(b,t)\}}$$

and

$$acc^{norm}(b,t) = \frac{acc(b,t)}{max\{acc(b,t)\}}$$

This allows the computation of a normalised vector:

$$o_{b,t}^{norm} = \left(div^{norm}(b,t), acc^{norm}(b,t)\right) \tag{3}$$

Gorthaur-Wheel then continues with the algorithm selection step using a roulette wheel method [13] which relies on the fitness $f_{b,t}$ of each algorithm. Namely, the fitness $f_{b,t}$ is obtained using the projection of the normalised vector $o_{b,t}^{norm}$ over the reference vector $\vec{c}$ (see Figure 1). It is thus possible to compute the fitness $f_{b,t}$ for each algorithm $b \in \mathcal{B}$ at iteration t as follows:

$$f_{b,t} = |o_{b,t}^{norm}| \cos \alpha_{b,t} - min_b\{|o_{b,t}^{norm}| \cos \alpha_{b,t}\} \tag{4}$$

Then, at each iteration t, *Gorthaur-Wheel* chooses an algorithm $b \in \mathcal{B}$ with a probability $p_{b,t}$ defined as follows:

$$p_{b,t} = \frac{f_{b,t} + \xi}{\sum_{i=1}^{|\mathcal{B}|} f_{b_i,t} + \xi} \tag{5}$$

where $\xi = 2^{-1074}$ is a constant which both avoids dividing by 0 and ensures that, at each step, each algorithm has a minimum probability of being selected by the roulette wheel.

The selection of arm/item $a \in A$ is then processed according to the strategy of algorithm b chosen by *Gorthaur-Wheel*. Finally, reward r_t is observed and algorithm b's parameters are updated (e.g., average reward of each arm, reward expectancy for *ε-Greedy* [30], response vector and covariance matrix in the case of *LinUCB* [1]).

In [13] it was observed that *Gorthaur-Wheel* selects algorithms (among its portfolio) which best fit the datasets or cases it encounters. This is a major advantage when dealing online with real-world applications. The performance of such an approach thus depends on both 1) the algorithms in the portfolio and 2) the strategy used to select an algorithm from the portfolio. Our first experiments with *Gorthaur-Wheel* related in [13] aimed at validating the portfolio approach implemented in *Gorthaur-Wheel* showed that in the case of recommendation, the method managed to cope with different types of datasets. Of course, *Gorthaur-Wheel* chose *CMAB* algorithms when faced with contextual settings and *MAB* algorithms for non-contextual problems. Also, with the datasets that it was experimented on, *Gorthaur-Wheel* was able to choose the algorithms that were most efficient in reaching the required accuracy/diversity target trade off. In this article we concentrate on the algorithm selection method. Here, rather than using a Roulette wheel selection based on each algorithm's fitness, we consider the selection of algorithms from the portfolio as a *MAB* problem and use the *EXP3 MAB* algorithm to solve it.

3.4. *Selection strategy:* EXP3

EXP3 stands for *EXPonential-weight algorithm for EXPloration and EXPloitation* [8, 31] and has been implemented to handle the non-stochastic adversarial multi-armed bandit problem. In this case, we assume that rewards obtained for the sequence generated by the Markov process are defined by an adversary (conscious or consciousless). *EXP3* operates by using reward estimates to produce a distribution over the arms. It basically maintains those estimations through a list of weights, one for each arm, and uses these weights to randomly select the arm to pull. Then, it receives a payoff for the chosen action (arm) and updates the weight with respect to this returned value of the payoff. Furthermore, in *EXP3* we introduce $\eta \in]0, 1]$ which is an egalitarianism factor aiming at adding more or less random selections (the closer η is to 1, the less the weights have an effect on the selection of the arms).

Namely, when using *EXP3*, the probability to choose arm a at iteration t is defined as follows:

$$P_{a,t} = (1 - \eta) \frac{w_{a,t}}{\sum_{i=1}^{k} w_{a_i,t}} + \frac{\eta}{k}$$

where $w_{a,t+1} = w_{a,t} \exp\left(\eta \, \frac{r_{a,t}}{p_{a,t} k}\right)$ if arm a was selected at iteration t and $r_{a,t}$ was the corresponding reward. Otherwise $w_{a,t+1} = w_{a,t}$. It has been proven for *EXP3* [8] that for a set of k arms, the regret upper bound $\rho_T \leq c \sqrt{kT \ln(k)}$ where c is a fixed constant.

One of the main advantages of using *EXP3* is that it continually explore sub-optimal actions. This property of *EXP3* should ensure robustness with respect to non-stationarity in cases where the model changes over time. Thus, even though *EXP3* generally obtains lower payoffs than other algorithms (see Table 1) due to its continuous exploration, it can adapt well to changing situations.

Since our new proposal for *Gorthaur-EXP3* uses an *EXP3* selection strategy, the *EXP3* algorithm is reminded in Algorithm 1.

3.5. *Background studies*

In order to determine the best competitive algorithms to include in *Gorthaur-EXP3*'s portfolio, we have carried out a preliminary study to evaluate several *MAB* and *CMAB* algorithms manually [11, 13, 32]. The results we obtained are presented in Table 1 and

Algorithm 1 - *EXP3* algorithm

Require: The set of k arms $a \in A$; horizon T; $\eta \in]0,1]$; $\forall a \in A, w_a = 1$.

for $t = 1$ to T **do**

  for $i = 1$ to k **do**

    Set $P_{a_i,t} = (1-\eta)\frac{w_{a_i,t}}{\sum_{j=1}^{k} w_{a_j,t}} + \frac{\eta}{k}$

  end for

  Select arm a_t randomly according to the probabilities $P_{a_1,t}, \ldots, P_{a_k,t}$

  Receive reward $r_{a_t,t}$

  for $j = 1$ to k **do**

$$\widehat{r}_{a_j,t} = \begin{cases} \frac{r_{a_j,t}}{p_{a_j,t}} & \text{if } a_j = a_t \\ 0 & \text{otherwise} \end{cases}$$

    Update $w_{a_j,t+1} = w_{a_j,t} \exp\left(\frac{\eta}{k}\,\widehat{r}_{a_j,t}\right)$

  end for

end for

will be considered as the reference when evaluating the results obtained by *Gorthaur-EXP3's* Bandit based selection strategy.

Furthermore, in our previous work [13] we noticed that *EXP4.P* [31] and *ε-Greedy* do not give any significant advantage to Gorthaur's portfolio compared to other algorithms. Moreover, we observed in [32] that a balance between the number of *MAB*s and of *CMAB*s in the portfolio helped reach a higher global efficiency (accuracy and diversity). Thus, in this work, we decide to remove *ε-Greedy* [30] and to replace *EXP4.P* by *SW-LinUCB* [11] in order to, both, better deal with non stationarity and increase diversity.

4. Method

4.1. Problem setting

In most real-world applications that recommend items to users, the criteria of global accuracy and diversity (See Section 3.3) are both important to consider [33] in order to make relevant recommendations and prevent users from being bored by the redundancy of similar recommendations. Thus, herein we state our problem as a bi-objective

	Global Accuracy	**Diversity**
Control	CMABs	CMABs
RS-ASM (ff)	LinUCB	SW-LinUCB
Food	LinUCB, LinTS	SW-LinUCB
RS-ASM (sf)	LinUCB, LinTS	SW-LinUCB
Movie Lens	LinUCB, LinTS	SW-LinUCB
Jester	UCB1, TS	EXP3
RS-ASM (season)	LinUCB, LinTS	SW-LinUCB
RS-ASM (LS10k-T)	UCB1, TS	EXP3
RS-ASM (LS10k-30k)	LinUCB	SW-LinUCB

Table 1: Best algorithms for each dataset and each criterion

optimization problem where the goal is to maximise the global accuracy and the diversity of recommendations made to users. Formally, it can be stated as follows [13]:

$$\max(div(t); acc(t)) \text{ s.t. } t \in [1, T]$$

In the case of our portfolio approach, at each time step t, one algorithm $b \in \mathcal{B}$ is selected from the portfolio such that:

$$b = \text{argmax}_{b_i \in \mathcal{B}}(div(b_i, t); acc(b_i, t))$$

Basically, our new method *Gorthaur-EXP3*, presented in the next section, uses a selection algorithm based on a *MAB* algorithm (*EXP3*) and is not able to manage a tuple as a reward but only a single value. Hence, this requires to aggregate both measures of accuracy and diversity. The level of aggregation that defines the balance of accuracy/diversity, is set by the value of Θ itself and can be either decided at the beginning [29] or dynamically computed [13]. Then, the reward update computation is carried out sequentially using the algorithm's fitness obtained from the projection of normalised vectors $o_{b_i,t}^{norm}$ over the reference vector $\vec{c}$ (determined by Θ).

4.2. Gorthaur-EXP3

The main difference between the original *Gorthaur-Wheel* method and the method proposed in this article is its algorithm selection strategy. At each iteration t, *Gorthaur-EXP3* chooses an algorithm $b_t \in \mathcal{B}$ with a probability $p_{b,t}$ defined as follows:

$$p_{b_t} = (1-\eta)\frac{w_{b_t}}{\sum_{j=1}^{|\mathcal{B}|} w_{b_{j,t}}} + \frac{\eta}{|\mathcal{B}|} \tag{6}$$

Gorthaur-EXP3 then selects an arm $a \in A$ according to the strategy of the chosen algorithm b_t and recommends the associated item to user u_t. Finally, *Gorthaur-EXP3* observes the obtained reward r_t and proceeds with the update of accuracy, diversity and fitness of algorithm b_t (See Equations 1 to 4) as the original *Gorthaur-Wheel* . Afterwards, for the chosen algorithm b_t *Gorthaur-EXP3* computes $\widehat{f}_{b,t} = \frac{f_{b,t}}{p_{b,t}}$ and $\widehat{f}_{b,t} = 0$ for others (i.e., algorithms that were not selected at trial t). Finally, *Gorthaur-EXP3* updates its weights as follows:

$$\forall b \in |\mathcal{B}|, \eta \in]0;1],\ w_{b,t+1} = w_{b,t}\ \exp\left(\frac{\eta}{|\mathcal{B}|}\ \widehat{f}_{b,t}\right) \tag{7}$$

A recommendation system using *Gorthaur-EXP3* works as shown in Algorithm 2.

4.3. Accuracy/Diversity Trade-off

In this article, we simulate real-world cases which require a balanced trade-off between accuracy and diversity. The settings for Algorithm 2 have been given with a fixed angle of $\Theta = \frac{\pi}{4}$ which corresponds to this balance. However, depending on application needs, it may be useful to favour either accuracy or diversity [13] therefore requiring different values of Θ. *Gorthaur-EXP3* determines the fitness of an algorithm b to reach the accuracy/diversity trade-off by computing the projection of the normalised vector $o_{b,t}^{norm}$ (See Equation 3) over the reference vector $\vec{c}$ (See Figure 1). Thus, the favoring granted to each criteria corresponds to the value of Θ itself (See Section 3.3[1]).

4.4. About expected regret

As proven by [23], when one uses a portfolio approach with reinforcement learning algorithms, the regret would not be worse than the worst algorithm. Moreover, accord-

[1]The reference vector $\vec{c}$ is defined by the angle $\Theta \in [0; \frac{\pi}{2}]$ made by $\vec{c}$ with the horizontal axis (ΔDiv)

Algorithm 2 - *Gorthaur-EXP3* Algorithm for recommendation

Require: List of users $u \in U$ and their context $x \in X$ (if context is available). List of k items to recommend associated to arms $a \in A$. The algorithms portfolio $b \in \mathcal{B}$. Angle $\Theta = 45°$. $\eta \in]0;1]$. $w_{b_{i,1}} = 1$ for $i = 1, .., |\mathcal{B}|$.

for $t = 1$ to T **do**

  Randomly select a user $u_t \in U$ and his context $x_t \in X$

  for all $b_i \in \mathcal{B}$ **do**

    Calculate $p_{b_{i,t}} = (1-\eta)\frac{w_{b_{i,t}}}{\sum_{j=1}^{|\mathcal{B}|} w_{b_{j,t}}} + \frac{\eta}{|\mathcal{B}|}$ as defined in Equation (6)

  end for

  Draw selection of algorithm b_t randomly according to the probabilities $p_{b_{1,t}}, \ldots, p_{b_{|\mathcal{B}|,t}}$, and then play algorithm b_t

  Select item $a \in A$ according to the strategy of the previously chosen algorithm b_t and recommend this item to user u_t

  Observe the obtained reward r_t

  Update parameters of the previously chosen algorithm b_t according to its reward processing strategy

  Update $acc(b_t, t)$ as defined in Equation (1)

  Update $div(b_t, t)$ as defined in Equation (2)

  Update $o_{b_t,t}^{norm}$ as defined in Equation (3)

  Update $f_{b_t,t}$ as defined in Equation (4)

  for $j = 1$ to $|\mathcal{B}|$ **do**

$$\widehat{f}_{b_j,t} = \begin{cases} \frac{f_{b_j,t}}{p_{b_j,t}} & \text{if } b_j = b_t \\ 0 & \text{otherwise} \end{cases}$$

    Update $w_{b_j,t+1} = w_{b_j,t} \ \exp\left(\eta \, \frac{\widehat{f}_{b_j,t}}{|\mathcal{B}|}\right)$

  end for

end for

---

ing to the no free lunch theorem it would also not be better than the best algorithm in the portfolio [34].

However, since the rewarding function fully relies on Gorthaur-EXP3's fitness calculation, we can ensure that the convergence of *Gorthaur-EXP3* is similar to that of the algorithm that best fits the target accuracy/diversity trade-off. Also, the final regret of *Gorthaur-EXP3* converges to that of its best algorithm proportionally to its probability of selection.

We expect *Gorthaur-EXP3* to be more selective than its previous version with Roulette Wheel selection.

Theorem 1. Gorthaur-EXP3 ***convergence of the accuracy/diversity trade-off in a stationary environment****.*
Let $\mathcal{B}$ be the set of algorithms in Gorthaur-EXP3's *portfolio and $b \in \mathcal{B}$ a specific algorithm of the portfolio where. Let $f_b(t)$ be the fitness of algorithm $b \in \mathcal{B}$ at iteration $t \in [1; T]$ where T is the finite horizon. Let $\eta \in]0; 1]$ be the egalitarianism factor to be initialized at time point $t_0 = 0$. Then, according to the general proof of convergence of* EXP3 *presented in [8] we can express that for* Gorthaur-EXP3 *we have:*
$\forall t \in [1; T]$; $\forall b \in \mathcal{B}$ *with* $|\mathcal{B}| > 0$; $\forall f_b(t)$; $\forall \eta \in]0; 1]$

$$\begin{aligned} G_{max,f}(T) - \mathbb{E}(G_{Gorthaur,f}(T)) \\ \leq (e-1)\eta G_{max,f}(T) + \frac{|\mathcal{B}| \ln(|\mathcal{B}|)}{\eta} \end{aligned}$$

where $G_{max,f}(T)$ is the maximum gain of fitness that can be obtained at horizon T by systematically selecting the optimal algorithm that best fits the accuracy/diversity trade-off and $G_{Gorthaur,f}(T)$ is the gain of fitness obtained with the Gorthaur-EXP3 *algorithm.*

The following section gives a sketch of the proof of convergence of the accuracy/diversity trade-off for *Gorthaur-EXP3*. It is based on the proof of convergence of the *EXP3* algorithm given in [8].

4.5. Gorthaur-EXP3 *accuracy/diversity trade-off analysis - Sketch of the proof*

By thoroughly following the general proof of *EXP3* [8] (Section 3), we can demonstrate the convergence of the *Gorthaur-EXP3* accuracy/diversity trade-off when using optimal algorithm selection (proof of Theorem 1) for a preset value of $\Theta = \frac{\pi}{4}$ corresponding to a balance between accuracy and diversity.

Proof (Sketch) of Theorem 1.

$\forall t \in [1;T]$; $\forall b \in \mathcal{B}$ with $|\mathcal{B}| > 0$ (to simplify further notations, let $z = |\mathcal{B}|$); $\forall f_b(t)$; $\forall \eta \in]0;1]$, let $W_t = \sum_{i=1}^{z} w_{b_i}(t)$ be the sum of each algorithm's weight. Note that $W_1 = z$ since $\forall b \in \mathcal{B}$ we have for $t = 1$, $w_b(1) = 1$ and $\forall b \in (B)$; $p_b(t) = (1-\eta)\frac{w_b(t)}{Wt} + \frac{\eta}{z}$ (See Algorithm 2).

For all sequence of selection algorithms drawn by *Gorthaur-EXP3*, we have:

$$\frac{W_{t+1}}{W_t} = \sum_{i=1}^{z} \frac{w_{b_i}(t)\ \exp\left(\frac{\eta}{z}\ \widehat{f}_{b_i}(t)\right)}{W_t},$$

$$= \sum_{i=1}^{z} \frac{p_{b_i}(t) - \frac{\eta}{z}}{1-\eta}\ \exp\left(\frac{\eta}{z}\ \widehat{f}_{b_i}(t)\right),$$

Since $\widehat{f}_b(t) \leq \frac{1}{p_b(t)} \leq \frac{z}{\eta}$, and $\exp(x) \leq 1 + x + (e-2)x^2$ when $x \leq 1$ we have:

$$\frac{W_{t+1}}{W_t} \leq \sum_{i=1}^{z} \frac{p_{b_i}(t) - \frac{\eta}{z}}{1-\eta}\ \left[1 + \frac{\eta}{z}\ \widehat{f}_{b_i}(t) + (e-2)\left(\frac{\eta}{z}\right)^2\ \widehat{f}_{b_i}(t)^2\right],$$

Since $\sum_{i=1}^{z} p_{b_i}(t)\ \widehat{f}_{b_i}(t) = f_b(t)$ and $\sum_{i=1}^{z} p_{b_i}(t)\ \widehat{f}_{b_i}(t)^2 \leq \sum_{i=1}^{z} \widehat{f}_{b_i}(t)$ we can write:

$$\frac{W_{t+1}}{W_t} \leq 1 + \frac{\left(\frac{\eta}{z}\right)}{1-\eta}\ f_{b_t}(t) + \frac{(e-2)\ \left(\frac{\eta}{z}\right)^2}{1-\eta}\ \sum_{i=1}^{z} \widehat{f}_{b_i}(t),$$

Considering that $1 + \Delta \leq \exp(\Delta)$ we have:

$$\ln\left(\frac{W_{t+1}}{W_t}\right) \leq \frac{\left(\frac{\eta}{z}\right)}{1-\eta}\ f_{b_t}(t) + \frac{(e-2)\ \left(\frac{\eta}{z}\right)^2}{1-\eta}\ \sum_{i=1}^{z} \widehat{f}_{b_i}(t),$$

When summing over t (operating with a finite horizon T), we get:

$$\ln\left(\frac{W_{T+1}}{W_1}\right) \leq \frac{\frac{\eta}{z}}{1-\eta}\ G_{Gorthaur,f}(T) + \frac{(e-2)\ \left(\frac{\eta}{z}\right)^2}{1-\eta}\ \sum_{i=1}^{z}\ \sum_{t=1}^{T} \widehat{f}_{b_i}(t),$$

For any algorithm selection b, we have:

$$\ln\left(\frac{W_{T+1}}{W_1}\right) \geq \frac{\ln\left(W_b(T+1)\right)}{\ln\left(W_1\right)} = \frac{\eta}{z}\ \sum_{t=1}^{T} \widehat{f}_b(t) - \ln(z),$$

we thus obtain the following equation:

$$G_{Gorthaur,f}(T) \geq (1-\eta)\ \left(\sum_{t=1}^{T} \widehat{f}_b(t) - \frac{z\ln(z)}{\eta}\right) - (e-2)\ \frac{\eta}{z}\ \sum_{i=1}^{z}\ \sum_{t=1}^{T} \widehat{f}_{b_i}(t),$$

When taking the expectation of both sides of the previous equation and since b is chosen arbitrarily such that $\sum_{t=1}^{T} f_{b_i}(t) \leq \eta\, G_{max}$ we have:

$$\mathbb{E}\left[G_{Gorthaur,f}(T)\right] \geq (1-\eta)\ \left(G_{max,f}(T) - \frac{z\ln(z)}{\eta}\right) - (e-2)\ \eta\ G_{max,f}(T),$$

and thus :

$$\mathbb{E}\left[G_{Gorthaur,f}(T)\right] - G_{max,f}(T) \geq -\eta\ (e-1)\ G_{max,f}(T) - \frac{z\ln(z)}{\eta},$$

which leads directly to Theorem 1:

$$G_{max,f}(T) - \mathbb{E}\left[G_{Gorthaur,f}(T)\right] \leq (e-1)\ \eta\ G_{max,f}(T) + \frac{|\mathcal{B}|\ln(|\mathcal{B}|)}{\eta}.$$

□

Once again, note that for more information about the complete analysis, both our sketch and theorem rely on the key article [8] and its detailed proof in Section 3.

Furthermore, since *Gorthaur-EXP3* selects the different algorithms of its portfolio according to their capacity to meet the desired trade-off, one can assume that at worst the final regret upper bound of *Gorthaur-EXP3* will be the sum of the regret upper bounds of the algorithms in the portfolio proportionally to their selection rate at Horizon T. However, according to the proof above, since *Gorthaur-EXP3* will process its exploitation step by selecting the most optimal algorithm from its portfolio, we argue that *Gorthaur-EXP3*'s final regret upper bound will tend towards that of its optimal algorithm.

4.6. Sharing rewards

In our previous works with *Gorthaur-Wheel* each algorithm of the portfolio learned from its own past recommendation history. The user's feedback on a recommendation resulting from a given algorithm only affected that specific algorithm. Here we would like to investigate whether reward sharing between algorithms of the portfolio can enhance the performances of all algorithms rather than those that are chosen. Even though the theoretical soundness of the approach still needs to be formally proven, experiments tend to show that it adapts well to cases where the reaction of users in given contexts is stable (stationary environments). Both approaches, with and without reward sharing between the algorithms of the portfolio, were considered in our experimental evaluation of *Gorthaur-EXP3*.

4.7. *Gorthaur's portfolio*

Following the different algorithms' criteria in terms of accuracy, personalization, diversity, and applicability to real-time applications [13], *Gorthaur-EXP3*'s portfolio is composed of the following 6 Multi-Armed Bandit algorithms both contextual and non contextual:

- **Multi-Armed Bandits algorithms (*MAB*):** *UCB1* [8], *Thompson Sampling* (*TS*) [16], and *EXP3* [8]
- **Contextual Multi-Armed Bandits algorithms (*CMAB*):** *LinUCB* [1], *SW-LinUCB* [11], and *Linear Thompson Sampling* (*LinTS*) [9].

In this article, a specific set of algorithms were selected according to particular needs. Nevertheless, it is possible to use other types of recommendation algorithms in *Gorthaur-EXP3*'s portfolio e.g., collaborative-filtering [35], Monte-Carlo Markov Chain [36] algorithms, etc.

5. Empirical evaluations

In this section, we first briefly remind the notion of context and describe the datasets we choose for our experiments (See Table 2) and the algorithms we compare. Finally, after detailing our experimental settings, we present and discuss our results.

5.1. *The notion of context*

Herein, for each dataset we make the context [1, 37, 38, 39, 40] exploitable by representing it by a features vector. Thus, the context is provided in a structured representation of binary variables (*one-hot encoding*). More precisely, its computation consists in transforming the continuous variables of the original datasets into categorical variables by dividing them according to their range following their quantiles. Then, we finally binarize ($\{0, 1\}$) the categorical variables in order to obtain a *one-hot* vector.

5.2. *Datasets*

Datasets have been chosen according to different criteria in terms of scale: Number of instances (from 424 to 1,025,010), number of binary features (from 0 to 270), and number of arms (from 4 to 100).

The evaluation of our proposal is based on five datasets (See Table 2) among which, one is an artificial dataset used for control, and four are real-world datasets:

Dataset name	Number of instances	Categorical features	Binary features	Number of arms	Dataset source
Control	1,000	4	4	4	Artificial
RS-ASM	2,152	8	56	18	Kaggle
Food	424	80	270	20	AIST
Movie Lens	943	43	51	1682	Grouplens
Jester	24,983	0	0	100	UC Berkeley

Table 2: Original Datasets

- **Control**: An artificially generated dataset with an equiproportional distribution between the four arms, and generated with a x^* which illustrates the optimal operation of a *CMAB* algorithm. As it was the case in our past contributions, this dataset is considered as the reference [11, 13].
- **Recommendation System for Angers Smart City (RS-ASM)** from Kaggle[2]: A dataset used both with full features (ff) and with sparse features (sf) by intentionally removing parts of the context (*preferences* and *hobbies*). This allows the observation of the impact of induced sparsity in this real-world dataset.

  The *RS-ASM* dataset has also been used for the non-stationary experiments and therefore modified according to the needs of the three different non-stationary cases we set: RS-ASM(season), RS-ASM(LS10k-T), and RS-AS(LS10k-30k) (see Simulation description in 5.3.1 for detailed explanations of modifications).
- **Food** from National Institute of Advanced Industrial Science and Technology (AIST), Japan: A context-aware food preference dataset for recommendation systems, used and distributed by Hideki Asoh [41].
- **Movie Lens 100K** from GroupLens Research[3]: A movie recommendation dataset

[2]https://www.kaggle.com/

[3]https://grouplens.org/datasets/movielens/100k/

in which 1682 movies are rated on a scale from 1 to 5 by 943 users.

- **Jester** from UC Berkeley[4]: A non-contextual dataset for joke recommendation.

RSASM dataset	**Environment specificities**	**Categorical features**	**Binary features**	**New dataset specificities**
(sf)	sparse context	6	24	Sparsity
(season)	non stationarity	8	56	Concept-shift
(LS10k-T)	non stationarity	$8 \rightarrow 0$	$56 \rightarrow 0$	Covariate-shift
(LS10k-30k)	non stationarity	$8 \rightarrow 0 \rightarrow 8$	$56 \rightarrow 0 \rightarrow 56$	Covariate-shift

Table 3: Special modifications on RS-ASM based Datasets

5.3. Experimental settings

5.3.1. Simulation

In order to simulate a data stream of arriving users with their contexts (see line 2 of Algorithm 2), we randomly select them sequentially from the whole dataset.

In the case of stationary experiments, since the number of instances is different between datasets, we need to scale up the time horizon T. Hence, depending on the size of the dataset, we set:

- 50,000 rounds for small datasets (i.e., with less than 2500 users), *Control*, *RS-ASM*, *Food*, and *Movie Lens*.

- 100,000 rounds for medium size dataset *Jester*.

Since learning in the presence of concept-drift or hidden contexts is an important challenge in the field of machine learning [39, 42], we focus our non-stationary and restricted context experiments on the observation of the effects of concept-drift/shift or covariate-shift. Thus, we construct five different cases from the RS-ASM dataset :

[4]http://eigentaste.berkeley.edu/dataset/

- A case of induced sparsity where context is given with missing features (See description of *RS-ASM (sf)* in 5.2).

- A case representing a change of season where the reward probabilities of the arms are changed after 25,000 rounds (*RS-ASM (season)*).

- A case where "sensors" are lost after 10,000 rounds and never recovered (*RS-ASM (LS10k-T)*).

- A case where "sensors" are lost after 10,000 rounds and recovered after 30,000 rounds (*RS-ASM (LS10k-30k)*).

The two latter cases are interesting to experiment since they correspond to situations in real-world application of mobile recommendation systems, smart cars or smart boats using recommendation systems where data from sensors can be partially or totally missing for some time.

5.3.2. Comparison of algorithms

Our new portfolio approach *Gorthaur-EXP3*, applying an *EXP3* algorithm selection is compared with the original *Gorthaur-Wheel* method that applies a *roulette wheel* algorithm selection [13].

All comparisons are made in terms of final values of global accuracy and diversity (as defined in equations 1 and 2 of sub-section 3.3). The convergence of both approaches up to horizon T and the speed at which they converge are also experimentally observed. The latter is an important criterion for real-world applications which do not only need a good asymptotic limit of global accuracy but should also ensure an acceptable time to reach it.

In our comparison and for both algorithm selection methods (*EXP3* and *roulette wheel*), two specific cases are considered : one in which rewards are shared between the algorithms of the portfolio and another in which they are not.

The above comparisons are made using datasets that correspond to both stationary and non-stationary environments.

In all cases, both implementations of Gorthaur are compared with the best and the worst algorithm of the portfolio.

Finally, concerning *Gorthaur-EXP3*'s setting, note that we set $\Theta = \frac{\pi}{4}$ i.e., $\Theta = 45°$, which we can consider as a *balanced* or *fair* parameterization between accuracy and diversity [13].

6. Results and discussion

This section presents the results obtained for all our simulations. The characteristics of each experiment are summarised in Table 4. The table also indicates, in column "*Presentation of Results*", where the detailed results of each experiment are presented.

Furthermore, a selection of results is presented graphically in Figures 2 and 3. They show the evolution of global accuracy over time and give an indication on the convergence of both *Gorthaur-Wheel* and *Gorthaur-EXP3* methods for the different datasets (some corresponding to stationary environments, others to non-stationary environments). We also plot the results of the best *CMAB* and *MAB* algorithms for comparison.

In the remaining part of this section, the evaluation metrics that we have used and the presentation that we have chosen for our results are first explained. Then, both algorithm selection methods (*Roulette weel* and *EXP3*) are compared in two different situations: stationary and non stationary environments. Finally, the effect of sharing rewards among the algorithms of the portfolio is discussed.

The main result of our experiments is that *Gorthaur-EXP3* outperforms *Gorthaur-Wheel* in all cases except for specific datasets that correspond to non-stationary co-variate shift environments (*RS-ASM (LS10K-30K)* and *RS-ASM (LS10K-T)*).

6.1. Evaluation metrics and presentation of results

Tables 5a to 8b, present the results obtained using each of *Gorthaur*'s selection methods (*Roulette wheel* and *EXP3*) and, for each experiment, the results obtained individually by each algorithm in the portfolio in terms of accuracy and diversity.

In each table, the first line labeled *"GORTHAUR-WHEEL"* or *"GORTHAUR-EXP3"* indicates the values of global accuracy and diversity[5] obtained with the given selection method. These global results are presented in the form ($acc(T)$ / $div(T)$). The following

[5]See Section 3.3 for a description of how to compute Accuracy and Diversity

Algorithm Selection	Reward sharing	Stationary / non-stationary	Presentation of Results
Wheel	No	Stationary	Table 5a
Wheel	Yes	Stationary	Table 6a
EXP3	No	Stationary	Table 5b
EXP3	Yes	Stationary	Table 6b
Wheel	No	non-stationary	Table 7a
EXP3	No	non-stationary	Table 7b
Wheel	Yes	non-stationary	Table 8a
EXP3	Yes	non-stationary	Table 8b

Table 4: Presentation of results

lines respectively indicate, for each algorithm of the portfolio: its proportion of use (by *Gorthaur*) as a percentage, its global accuracy, and its diversity. These figures are presented in the form $(\text{proportion in } \% \ / \ acc(T) \ / \ div(T))$. Moreover, note that the value of Θ given both in radians and in degrees indicates the trade-off set between accuracy and diversity. Depending on the requirements of the real-world application, Θ can be set to favour either accuracy (i.e., $\Theta \to \frac{\pi}{2}$) or diversity (i.e., $\Theta \to 0$) [13]. Note that Θ can also be set to be dynamically self-calculated in order to determine the *Pareto* front of both criteria according to the algorithms in the portfolio [13]. Here, we decide to use a value of $\Theta = \frac{\pi}{4}$ indicating a balance between accuracy and diversity. The idea is to simulate a real-world recommendation system in which diversity of recommendation is as important as accuracy.

6.2. Gorthaur-EXP3 *Versus* Gorthaur-Wheel

Here, the results obtained by both algorithm selection methods are presented and discussed. Two different experimental settings are considered; The first corresponding to a stationary environment and the other to a non-stationary environment.

6.2.1. *Evaluation in a stationary environment*

The results obtained when running the experiments under stationary conditions are presented in Tables 5 and 6, and in Figure 2. They show that for all datasets except for the non contextual one (i.e., *Jester*), *Gorthaur-EXP3* outperforms *Gorthaur-Wheel* both in terms of final global accuracy and in terms of diversity.

	Control	RS-ASM (ff)	Food	Jester	Movie Lens
GORTHAUR-WHEEL	0.78 / 0.92	0.54 / 0.76	0.82 / 0.78	0.55 / 0.4	3.59 / 0.9
LinUCB	**23.9% / 0.99 / 0.99**	**22.1% / 0.59 / 0.88**	**21.5% / 0.89 / 0.91**	4% / 0.3 / 0.99	**17.7% / 3.56 / 0.96**
LinTS	**24.5% / 0.99 / 0.99**	19.8% / 0.49 / 0.88	20.4% / 0.86 / 0.86	3.9% / 0.3 / 0.99	16.9% / 3.67 / 0.86
SW-LinUCB	**24.5% / 0.98 / 0.99**	**22% / 0.58 / 0.9**	**21.3% / 0.87 / 0.94**	4.4% / 0.3 / 0.99	**17.5% / 3.56 / 0.97**
UCB1	5.5% / 0.25 / 0.01	8.1% / 0.5 / 0.31	7% / 0.65/ 0.02	27% / 0.67 / 0.28	13% / 3.87 / 0.32
EXP3	12.2% / 0.25 / 0.69	16.4% / 0.5 / 0.55	16.9% / 0.7 / 0.6	**36.5% / 0.45 / 0.78**	17.2% / 3.4 / 0.96
TS	10.2% / 0.24 / 0.6	11.6% / 0.57 / 0.17	12.9% / 0.78 / 0.22	24.2% / 0.56 / 0.4	**17.7% / 3.57 / 0.93**

(a) *Gorthaur-Wheel* ($\Theta = \frac{\pi}{4}$, i.e., $\Theta = 45°$)

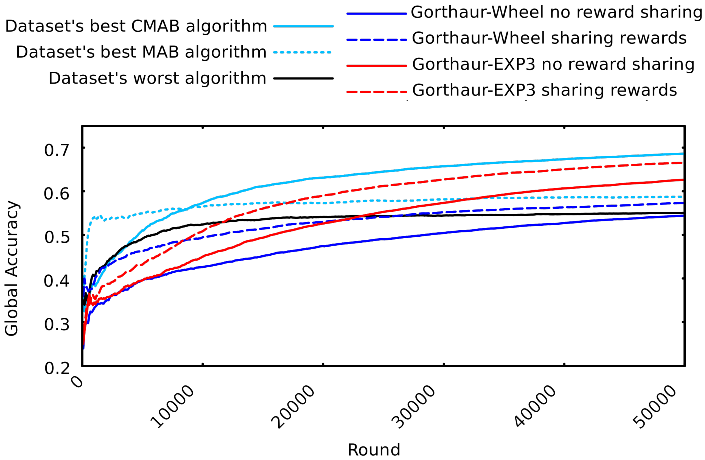
	Control	RS-ASM (ff)	Food	Jester	Movie Lens
GORTHAUR-EXP3	0.97 / 0.99	0.63 / 0.86	0.9 / 0.92	0.46 / 0.52	3.83 / 0.91
LinUCB	**70.9% / 0.99 / 0.99**	**71.8% / 0.67 / 0.85**	20.5% / 0.88 / 0.91	1.4% / 0.3 / 0.99	**76.2% / 3.96 / 0.87**
LinTS	1.5% / 0.96 / 0.96	5.6% / 0.4 / 0.93	1.8% / 0.69 / 0.94	1.4% / 0.3 / 0.99	4.5% / 3.32 / 0.96
SW-LinUCB	24.5% / 0.97 / 0.99	17.6% / 0.54 / 0.9	**71.1% / 0.93 / 0.93**	1.3% / 0.3 / 0.99	4.9% / 3.34 / 0.98
UCB1	0.8% / 0.25 / 0.04	1.9% / 0.55 / 0.03	2% / 0.63 / 0.14	2% / 0.61 / 0.3	8.8% / 3.64 / 0.76
EXP3	1.5% / 0.26 / 0.76	2% / 0.35 / 0.9	2.2% / 0.62 / 0.89	**93.5% / 0.51 / 0.52**	2.3% / 3.29 / 0.94
TS	0.8% / 0.23 / 0.67	1% / 0.47 / 0.56	2.4% / 0.71 / 0.49	0.4% / 0.36 / 0.4	3.3% / 3.29 / 0.94

(b) *Gorthaur-EXP3* ($\Theta = \frac{\pi}{4}$, i.e., $\Theta = 45°$)

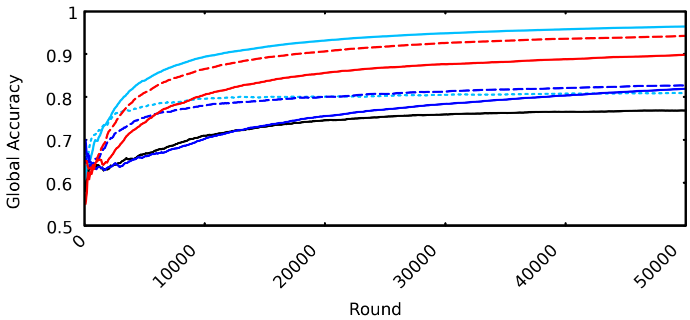
Table 5: Results with stationary environment and no reward sharing

With regards to the proportion of selection of each algorithm, the main difference between both selection methods is that *Gorthaur-Wheel* selects algorithms proportionally to their ability to best fit the accuracy/diversity trade-off, whereas *Gorthaur-EXP3* works as a bandit algorithm and eventually selects the algorithm which obtains the best trade-off. This difference is important to notice since the chosen selection strategy determines the result we want to really obtain in the real-world application. Namely, the question we need to answer is: Do we want to find the algorithm that best fits the targeted accuracy/diversity trade-off or an optimal proportion of several algorithms in the portfolio that can reach this trade-off ?

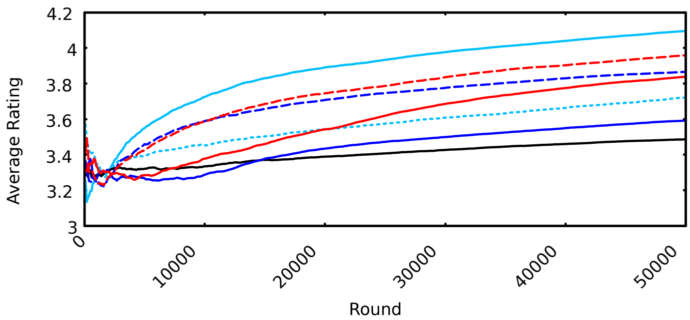
This can explain why, in most cases, *Gorthaur-EXP3* performs better than *Gorthaur-Wheel* according to both criteria.



(a) *RS-ASM (ff)*



(b) *Food*



(c) *Movie Lens*

Figure 2: Evolution of global accuracy with stationary environment

	Control	RS-ASM (ff)	Food	Jester	Movie Lens
GORTHAUR-WHEEL	0.78 / 0.99	0.57 / 0.79	0.84 / 0.89	0.49 / 0.32	3.86 / 0.89
LinUCB	**23.6% / 0.99 / 0.99**	**21.2% / 0.65 / 0.88**	**20.2% / 0.95 / 0.95**	6.8% / 0.3 / 0.99	**17.6% / 3.73 / 0.95**
LinTS	**23.6% / 0.99 / 0.99**	19.9% / 0.62 / 0.79	19.2% / 0.94 / 0.87	6.8% / 0.3 / 0.99	17.3% / 3.92 / 0.9
SW-LinUCB	**23.6% / 0.99 / 0.99**	**20.4% / 0.6 / 0.9**	**19.9% / 0.9 / 0.95**	6.8% / 0.3 / 0.99	**17.6% / 3.7 / 0.96**
UCB1	10.5% / 0.24 / 0.89	12.9% / 0.53 / 0.51	14.4% / 0.66 / 0.84	12.5% / 0.44 / 0.4	13% / 4.22 / 0.55
EXP3	8.1% / 0.25 / 0.74	15.4% / 0.41 / 0.78	15.6% / 0.64 / 0.89	**57.1% / 0.51 / 0.4**	17.3% / 3.75 / 0.9
TS	10.6% / 0.26 / 0.9	10.2% / 0.57 / 0.28	10.7% / 0.75 / 0.48	10% / 0.54 / 0.2	17.2% / 3.95 / 0.81

(a) *Gorthaur-Wheel* ($\Theta = \frac{\pi}{4}$, i.e., $\Theta = 45°$)

	Control	RS-ASM (ff)	Food	Jester	Movie Lens
GORTHAUR-EXP3	0.97 / 0.99	0.67 / 0.87	0.94 / 0.9	0.47 / 0.56	3.95 / 0.9
LinUCB	**74.3% / 0.99 / 0.99**	**80.3% / 0.71 / 0.87**	**90.9% / 0.96 / 0.9**	1.7% / 0.3 / 0.99	**79.2% / 4 / 0.9**
LinTS	10.3% / 0.99 / 0.97	10.1% / 0.53 / 0.86	2.5% / 0.9 / 0.85	1.7% / 0.3 / 0.99	7.9% / 3.73 / 0.92
SW-LinUCB	11.7% / 0.99 / 0.98	2.8% / 0.51 / 0.95	1.8% / 0.86 / 0.95	1.7% / 0.3 / 0.99	2% / 3.6 / 0.95
UCB1	1% / 0.26 / 0.88	1.5% / 0.46 / 0.43	1.5% / 0.65 / 0.73	0.6% / 0.35 / 0.53	2.6% / 3.76 / 0.51
EXP3	0.9% / 0.26 / 0.69	4.3% / 0.38 / 0.85	2.3% / 0.65 / 0.9	**93.7% / 0.51 / 0.58**	6.8% / 3.71 / 0.88
TS	1.8% / 0.26 / 0.93	1% / 0.4 / 0.5	1% / 0.73 / 0.54	0.6% / 0.3 / 0.36	1.5% / 3.84 / 0.8

(b) *Gorthaur-EXP3* ($\Theta = \frac{\pi}{4}$, i.e., $\Theta = 45°$)

Table 6: Results with stationary environment and sharing rewards

However, for the experiment using the non-contextual *Jester* dataset, *Gorthaur-EXP3* obtains better results than *Gorthaur-Wheel* in terms of diversity despite being eventually less accurate. This can be explained by the performance of the optimal algorithm selected by *Gorthaur-EXP3* which is chosen in terms of accuracy/diversity trade-off. In this specific case, the optimal algorithm chosen from the portfolio is *EXP3*. Out of all the algorithms of the portfolio, it is the one that offers the best trade-off between both criteria - i.e. for which the projection of the (Acc, Div) vector on the reference vector defined by Θ has the highest norm (*UCB1* and *TS* reach higher levels of global accuracy, but remain lower in terms of diversity).

On Figure 2, we observe that in cases where rewards are shared,*Gorthaur-Wheel* outperforms *Gorthaur-EXP3* when horizon T is small (i.e., $T < 10000$). This can be explained by the fact that *Gorthaur-EXP3* uses a *EXP3* selection strategy which involves an early and continuing exploration step. During this period, the selection strategy does not solely exploit the optimal algorithm unlike the Roulette Wheel selection which rapidly chooses among the set of optimal algorithms proportionally to their fitnesses. We do not observe this particular result when rewards are not shared, because algorithms among the portfolio do not stay in competition for long since they do not share their knowledge.

6.2.2. *Evaluation in a non-stationary environment*

The results obtained when running our experiments under non-stationary conditions are presented in Tables 7 and 8, and Figure 3.

	RS-ASM (sf)	RS-ASM (season)	RS-ASM (LS10k-T)	RS-ASM (LS10k-30k)
GORTHAUR-WHEEL	0.54 / 0.58	0.53 / 0.77	0.41 / 0.56	0.45 / 0.67
LinUCB	**20.9% / 0.56 / 0.6**	**21.4% / 0.58 / 0.89**	16% / 0.27 / 0.25	17.5% / 0.4 / 0.61
LinTS	19.7% / 0.56 / 0.55	**20.6% / 0.5 / 0.87**	15.7% / 0.26 / 0.24	17.3% / 0.37 / 0.62
SW-LinUCB	**21.7% / 0.51 / 0.74**	**21.6% / 0.56 / 0.91**	15% / 0.27 / 0.24	17.5% / 0.38 / 0.61
UCB1	7.3% / 0.54 / 0.01	6.5% / 0.3 / 0.03	9.8% / 0.56 / 0.01	9.6% / 0.45 / 0.29
EXP3	18.2% / 0.53 / 0.47	17.2% / 0.52 / 0.57	**25.7% / 0.51 / 0.54**	**21.7% / 0.52 / 0.45**
TS	12.2% / 0.58 / 0.13	12.7% / 0.57 / 0.17	17.8% / 0.57 / 0.18	16.4% / 0.57 / 0.22

(a) *Gorthaur-Wheel* ($\Theta = \frac{\pi}{4}$, i.e., $\Theta = 45°$)

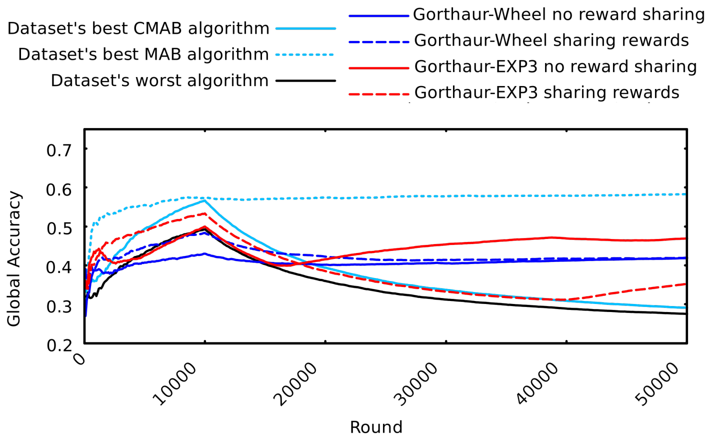
	RS-ASM (sf)	RS-ASM (season)	RS-ASM (LS10k-T)	RS-ASM (LS10k-30k)
GORTHAUR-EXP3	0.57 / 0.59	0.65 / 0.87	0.47 / 0.57	0.54 / 0.74
LinUCB	**68.8% / 0.6 / 0.51**	**88.8% / 0.68 / 0.87**	26.2% / 0.35 / 0.45	**59.2% / 0.57 / 0.8**
LinTS	5% / 0.49 / 0.67	2.5% / 0.36 / 0.95	1.9% / 0.22 / 0.22	4% / 0.3 / 0.66
SW-LinUCB	21.3% / 0.51 / 0.74	1.4% / 0.36 / 0.97	6.1% / 0.22 / 0.22	1.5% / 0.3 / 0.6
UCB1	1% / 0.22/ 0.19	1.7% / 0.53 / 0.11	1.9% / 0.49 / 0.37	1.9% / 0.57 / 0.06
EXP3	3.2% / 0.44 / 0.71	4.2% / 0.43 / 0.83	**60.4% / 0.55 / 0.57**	32% / 0.52 / 0.46
TS	0.7% / 0.5 / 0.5	1% / 0.49 / 0.44	3.5% / 0.52 / 0.56	1.4% / 0.47 / 0.68

(b) *Gorthaur-EXP3* ($\Theta = \frac{\pi}{4}$, i.e., $\Theta = 45°$)

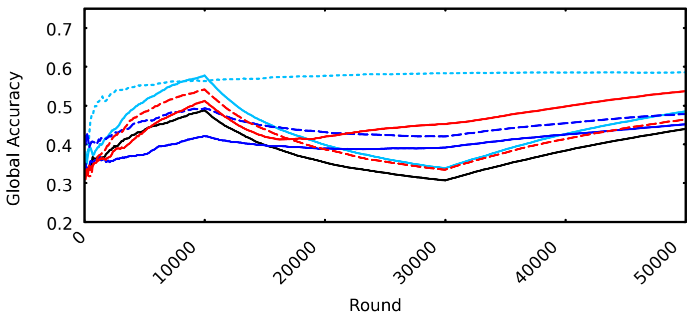
Table 7: Results with non-stationary environments and no reward sharing

From these results, we observe that in most cases *Gorthaur-EXP3* outperforms *Gorthaur-Wheel*. This is the case for all experiments in which rewards are not shared among the algorithms of the portfolio (Table 7).

However, in cases where rewards are shared, we observe in Table 8, and in Figure 3 that for *RS-ASM (10k-T)* and *RS-ASM (10k-30k)* datasets (i.e., covariate-shift cases) *Gorthaur-Wheel* outperforms *Gorthaur-EXP3* both in terms of final global accuracy and diversity. Nevertheless, we observe in Tables 8a and 8b that for *RS-ASM (sf)* and *RS-ASM (season)* datasets (i.e., concept-drift/shift cases) *Gorthaur-EXP3* outperforms *Gorthaur-Wheel* both in terms of final global accuracy and diversity.



(a) *RS-ASM (10k-T)*



(b) *RS-ASM (10k-30k)*

Figure 3: Evolution of global accuracy with non-stationary environment

Thus in the case when rewards are shared among algorithms, a roulette-wheel selection strategy allows to best deal with covariate-shift cases while it is preferable to use an *EXP3* strategy when faced with concept-drift/shift. However, in real-world applications, it is difficult to predict when and whether concept-drift/shift or covariate-shift may occur. When one needs to apply Gorthaur in real recommendation system applications, we advocate that to deal with non-stationarity, the possible risks of both concept-drift and covariate-shift should be studied before choosing the relevant selection strategy.

	RS-ASM (sf)	RS-ASM (season)	RS-ASM (LS10k-T)	RS-ASM (LS10k-30k)
GORTHAUR-WHEEL	0.56 / 0.56	0.58 / 0.78	0.42 / 0.48	0.48 / 0.67
LinUCB	19% / 0.59 / 0.55	**20.6% / 0.66 / 0.86**	14.5% / 0.27 / 0.24	17.1% / 0.45 / 0.63
LinTS	18.4% / 0.62 / 0.49	**19.9% / 0.62 / 0.77**	15.4% / 0.28 / 0.25	16.9% / 0.44 / 0.62
SW-LinUCB	**21.1% / 0.5 / 0.76**	**20.2% / 0.58 / 0.92**	14.6% / 0.28 / 0.25	16.5% / 0.43 / 0.62
UCB1	12.8% / 0.58 / 0.24	13.1% / 0.54 / 0.55	18.4% / 0.58 / 0.25	17% / 0.55 / 0.45
EXP3	16.8% / 0.52 / 0.56	15.4% / 0.44 / 0.77	**22.6% / 0.46 / 0.45**	**20.2% / 0.44 / 0.63**
TS	11.9% / 0.58 / 0.25	10.8% / 0.56 / 0.38	14.5% / 0.58 / 0.58	12.3% / 0.58 / 0.16

(a) *Gorthaur-Wheel*($\Theta = \frac{\pi}{4}$, i.e., $\Theta = 45°$)

	RS-ASM (sf)	RS-ASM (season)	RS-ASM (LS10k-T)	RS-ASM (LS10k-30k)
GORTHAUR-EXP3	0.6 / 0.57	0.67 / 0.84	0.35 / 0.4	0.46 / 0.62
LinUCB	**89.8% / 0.62 / 0.57**	**72.9% / 0.71 / 0.82**	**50.6% / 0.3 / 0.27**	2.2% / 0.46 / 0.56
LinTS	3.1% / 0.55 / 0.56	3.1% / 0.64 / 0.78	13.7% / 0.3 / 0.27	9.6% / 0.34 / 0.43
SW-LinUCB	1.2% / 0.4 / 0.83	18.9% / 0.61 / 0.91	3.2% / 0.23 / 0.24	**81.8% / 0.48 / 0.63**
UCB1	1.4% / 0.5 / 0.55	1.7% / 0.48 / 0.47	1.8% / 0.55 / 0.28	1.9% / 0.48 / 0.49
EXP3	3.5% / 0.51 / 0.56	2.2% / 0.4 / 0.84	29.5% / 0.44 / 0.42	3.2% / 0.31 / 0.52
TS	1% / 0.55 / 0.33	1.2% / 0.47 / 0.59	1.2% / 0.61 / 0.19	1.3% / 0.49 / 0.55

(b) *Gorthaur-EXP3* ($\Theta = \frac{\pi}{4}$, i.e., $\Theta = 45°$)

Table 8: Results with non-stationary environment and sharing rewards

6.3. Sharing Versus Not sharing rewards

The results presented in Tables 5 and 6 show that in stationary environments and when dealing with contextual datasets, the sharing of rewards among the algorithms of the portfolio always increases the final global accuracy whatever the algorithm selection method (*Wheel* or *EXP3*). In most cases, reward sharing seems to also increase the value of diversity. It is the case with all contextual datasets except for *MovieLens* (with both algorithm selection methods) and for *Food* when using the *EXP3* algorithm selection method. In those specific cases, the sharing of rewards very slightly decreases the final value of diversity. For the non contextual dataset (*Jester*, only *Gorthaur-EXP3* seems to benefit from reward sharing. When applied with *Gorthaur-Wheel*, reward sharing fails to increase both the final global accuracy and the diversity of the proposed recommendations.

In non-stationary experimental settings, we observe in Tables 8 and Figure 3 that in the case of concept-drift/shift datasets, sharing rewards benefits to both methods in terms of final global accuracy. Note that it does not improve diversity which stays similar or is up to 3,4% lower in the case where rewards are shared.

However, in the case of covariate-shift datasets, sharing rewards benefits in terms of final global accuracy to the roulette wheel selection strategy but not to the *EXP3* strategy which totally under-performs compared to the non sharing case. We assume that when algorithms share rewards, even though *EXP3* eventually discovers the new optimal algorithm to select, it remains unable to re-compute the right probability selection rapidly enough. Contrarily, when rewards are not shared, *EXP3*'s continuous exploration ability allows it to rapidly consider and recover the new optimal algorithm to use.

7. Conclusion and perspectives

In this article, we propose *Gorthaur-EXP3*: a novel portfolio approach of *MAB* and *CMAB* algorithms for recommendation which extends the original *Gorthaur-Wheel* method [13]. It aims at finding , among a portfolio, the algorithm which best maximises both criteria of global accuracy and diversity of the recommendations made to users.

More generally, *Gorthaur-EXP3* selects, from a given set of algorithms, the optimal algorithm that best fits the datasets or cases it faces. We argue that online real-world applications can benefit from such a method which would give an essential advantage by automatically and rapidly identifying the algorithm to use in different cases. Moreover, this identification relies on the *EXP3* guarantees that were previously theoretically proven. This gives strong confidence that the algorithm finds the optimal algorithm among a portfolio. Furthermore, *EXP3* has a continuous exploration mechanism that ensures its robustness in non stationary conditions which are typically encountered in various online applications.

In this article, we observe that *Gorthaur-EXP3* outperforms the original *Gorthaur-Wheel* method in most cases except in two specific cases: 1) when evaluations are conduced in a context-free and stationary environment and rewards are not shared; 2) when evaluations are carried out under non-stationary conditions due to covariate-shift (in simulation cases where the non-stationary conditions are due to concept-drift/shift *Gorthaur-EXP3* remains better).

In the Mobile Crowd Sensing and Computing paradigm, we are supposed to rely on rich context features sensed by user devices. Moreover, in such real world applications we consider that the risk of covariate-shift is low, predictable and possibly mitigable. Therefore, based on the results that we have obtained, we strongly believe that using *Gorthaur-EXP3* instead of the original *Gorthaur-Wheel* method is the best choice to make.

In future works, it will be interesting to:

1. Add more criteria to the optimization problem (e.g., individual accuracy) with which *Gorthaur-EXP3* will need to operate in a sphere or a n-sphere. This multi-objective perspective can be the centerpiece of multiple online applications which may require to optimise multiple criteria (e.g., sailing, cultural and social mobile recommendation systems, group recommendations).
2. Integrate and evaluate *Gorthaur-EXP3* in an online application in order to study real world aspects (e.g., real non stationary cases, partial user feedbacks) that cannot be fully observed when evaluating offline.

Both perspectives still remain to be experimented and will thus, naturally be our next step.

References

[1] L. Li, W. Chu, J. Langford, R. E. Schapire, A contextual-bandit approach to personalized news article recommendation, in: Proceedings of the 19th international conference on World wide web, ACM, 2010, pp. 661–670. doi:10.1145/1772690.1772758.

[2] D. Bouneffouf, A. Bouzeghoub, A. L. Gançarski, A contextual-bandit algorithm for mobile context-aware recommender system, in: International Conference on Neural Information Processing, Springer, 2012, pp. 324–331. doi:10.1007/978-3-642-34487-9_40.

[3] P. Kohli, M. Salek, G. Stoddard, A fast bandit algorithm for recommendation to users with heterogenous tastes, in: Twenty-Seventh AAAI Conference on Artificial Intelligence, 2013, pp. 1135–1141. doi:10.5555/2891460.2891618.

[4] J. Mary, R. Gaudel, P. Preux, Bandits and recommender systems, in: International Workshop on Machine Learning, Optimization and Big Data, Springer, 2015, pp. 325–336. doi:10.1007/978-3-319-27926-8_29.

[5] C. Zeng, Q. Wang, S. Mokhtari, T. Li, Online context-aware recommendation with time varying multi-armed bandit, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 2025–2034. doi:10.1145/2939672.2939878.

[6] J. Langford, T. Zhang, The epoch-greedy algorithm for multi-armed bandits with side information, in: Advances in neural information processing systems, 2008, pp. 817–824. doi:10.5555/2981562.2981665.

[7] H. Robbins, Some aspects of the sequential design of experiments, Bulletin of the American Mathematical Society (1952) 527–535.

[8] P. Auer, N. Cesa-Bianchi, Y. Freund, R. E. Schapire, The nonstochastic multiarmed bandit problem, SIAM journal on computing 32 (1) (2002) 48–77. doi:10.1137/S0097539701398375.

[9] S. Agrawal, N. Goyal, Thompson sampling for contextual bandits with linear payoffs, in: International Conference on Machine Learning, 2013, pp. 127–135. doi:10.5555/3042817.3043073.

[10] B. Brodén, M. Hammar, B. J. Nilsson, D. Paraschakis, Ensemble recommendations via thompson sampling: an experimental study within e-commerce, in: 23rd International Conference on Intelligent User Interfaces, 2018, pp. 19–29. doi:10.1145/3172944.3172967.

[11] N. Gutowski, T. Amghar, O. Camp, F. Chhel, Global versus individual accuracy in contextual multi-armed bandit, in: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, 2019, pp. 1647–1654. doi:10.1145/3297280.3297440.

[12] J. Kleinberg, S. Mullainathan, M. Raghavan, Inherent trade-offs in the fair determination of risk scores, arXiv preprint arXiv:1609.05807 (2016).

[13] N. Gutowski, T. Amghar, O. Camp, F. Chhel, Gorthaur: A portfolio approach for dynamic selection of multi-armed bandit algorithms for recommendation, in: 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), IEEE, 2019, pp. 1164–1171. doi:10.1109/ICTAI.2019.00161.

[14] P. Auer, Using confidence bounds for exploitation-exploration trade-offs, Journal of Machine Learning Research 3 (Nov) (2002) 397–422. doi:10.5555/944919.944941.

[15] D. Bouneffouf, R. Feraud, Multi-armed bandit problem with known trend, Neurocomputing 205 (2016) 16–21. doi:j.neucom.2016.02.052.

[16] S. Agrawal, N. Goyal, Analysis of Thompson sampling for the multi-armed bandit problem, in: Conference on Learning Theory, 2012, pp. 39–1.

[17] J. R. Rice, et al., The algorithm selection problem, Advances in computers 15 (65-118) (1976) 5. doi:10.1016/S0065-2458(08)60520-3.

[18] L. Kotthoff, Algorithm selection for combinatorial search problems: A survey, in: Data Mining and Constraint Programming, Springer, 2016, pp. 149–190. doi:10.1007/978-3-319-50137-6_7.

[19] L. Xu, F. Hutter, H. H. Hoos, K. Leyton-Brown, Satzilla: Portfolio-based algorithm selection for sat, Journal of artificial intelligence research 32 (1) (2008) 565–606. doi:10.5555/1622673.1622687.

[20] A. Fialho, L. Costa, M. Schoenauer, M. Sebag, Analyzing bandit-based adaptive operator selection mechanisms, Annals of Mathematics and Artificial Intelligence 60 (2010) 25–64. doi:10.1007/s10472-010-9213-y.

[21] M. Gagliolo, J. Schmidhuber, Learning dynamic algorithm portfolios, Annals of Mathematics and Artificial Intelligence 47 (3-4) (2006) 295–328. doi:10.1007/s10472-006-9036-z.

[22] K. A. Smith-Miles, Cross-disciplinary perspectives on meta-learning for algorithm selection, ACM Computing Surveys (CSUR) 41 (1) (2009) 1–25. doi:10.1145/1456650.1456656.

[23] R. Laroche, R. Feraud, Reinforcement learning algorithm selection, in: International Conference on

Learning Representations, 2018.

[24] R. Allesiardo, R. Féraud, Selection of learning experts, in: International Joint Conference on Neural Networks (IJCNN), IEEE, 2017, pp. 1005–1010. doi:10.1109/IJCNN.2017.7965962.

[25] R. Busa-Fekete, B. Szörényi, P. Weng, S. Mannor, Multi-objective bandits: optimizing the generalized gini index, in: Proceedings of the 34th International Conference on Machine Learning, JMLR, 2017, pp. 625–634. doi:10.5555/3305381.3305446.

[26] M. M. Drugan, A. Nowe, Designing multi-objective multi-armed bandits algorithms: A study, in: The International Joint Conference on Neural Networks (IJCNN), IEEE, 2013, pp. 1–8. doi:10.1109/IJCNN.2013.6707036.

[27] A. Lacerda, Multi-objective ranked bandits for recommender systems, Neurocomputing 246 (2017) 12–24. doi:10.1016/j.neucom.2016.12.076.

[28] C. Tekin, E. Turğay, Multi-objective contextual multi-armed bandit with a dominant objective, IEEE Transactions on Signal Processing 66 (14) (2018) 3799–3813. doi:10.1109/MLSP.2017.8168123.

[29] J. Maturana, F. Saubion, A compass to guide genetic algorithms, in: International Conference on Parallel Problem Solving from Nature, Springer, 2008, pp. 256–265. doi:10.5555/2951659.2951687.

[30] R. S. Sutton, A. G. Barto, et al., Introduction to reinforcement learning, Vol. 135, MIT press Cambridge, 1998.

[31] P. Auer, N. Cesa-Bianchi, Y. Freund, R. E. Schapire, Gambling in a rigged casino: The adversarial multi-armed bandit problem, in: Proceedings of IEEE 36th Annual Foundations of Computer Science, IEEE, 1995, pp. 322–331. doi:10.1109/SFCS.1995.492488.

[32] N. Gutowski, Context-aware recommendation systems for cultural events recommendation in Smart Cities, Thesis, Université d'Angers, France (Nov. 2019).

[33] S. M. McNee, J. Riedl, J. A. Konstan, Being accurate is not enough: how accuracy metrics have hurt recommender systems, in: international Conference of Human-Computer Interaction., 2006, pp. 1097–1101. doi:10.1145/1125451.1125659.

[34] Y.-C. Ho, D. L. Pepyne, Simple explanation of the no-free-lunch theorem and its implications, Journal of optimization theory and applications 115 (3) (2002) 549–570. doi:10.1109/CDC.2001.980896.

[35] F. Ricci, L. Rokach, B. Shapira, Recommender systems: introduction and challenges, in: Recommender systems handbook, Springer, 2015, pp. 1–34. doi:10.1007/978-1-4899-7637-6_1.

[36] A. Ansari, S. Essegaier, R. Kohli, Internet recommendation systems (2000).

[37] P. Brézillon, Context in artificial intelligence: A survey of the literature, Computers and artificial intelligence 18 (1999) 321–340. doi:10.1017/S0269888999141018.

[38] A. K. Dey, Understanding and using context, Personal Ubiquitous Computing 5 (1) (2001) 4–7. doi:10.1007/s007790170019.

[39] D. Bouneffouf, I. Rish, G. A. Cecchi, R. Féraud, Context attentive bandits: contextual bandit with restricted context, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence, 2017, pp. 1468–1475. doi:10.5555/3172077.3172091.

[40] N. Gutowski, T. Amghar, O. Camp, F. Chhel, Context enhancement for linear contextual multi-armed bandits, in: 2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI), IEEE,

2018, pp. 1048–1055. `doi:10.1109/ICTAI.2018.00161`.

[41] C. Ono, Y. Takishima, Y. Motomura, H. Asoh, Context-aware preference model based on a study of difference between real and supposed situation data, User Modeling, Adaptation, and Personalization (2009) 102–113.

[42] G. Widmer, M. Kubat, Learning in the presence of concept drift and hidden contexts, Machine learning 23 (1) (1996) 69–101.