Generic Exploration and K-armed Voting Bandits (extended version)

Tanguy Urvoy TANGUY.URVOY@ORANGE.COM
Fabrice Clerot FABRICE.CLEROT@ORANGE.COM
Raphael Féraud RAPHAEL.FERAUD@ORANGE.COM
Sami Naamane SAMI.NAAMANE@ORANGE.COM
Orange-labs, 2 avenue Pierre Marzin, 22307 Lannion, FRANCE

Abstract

We study a stochastic online learning scheme with partial feedback where the utility of decisions is only observable through an estimation of the environment parameters. We propose a generic pure-exploration algorithm, able to cope with various utility functions from multi-armed bandits settings to dueling bandits. The primary application of this setting is to offer a natural generalization of dueling bandits for situations where the environment parameters reflect the idiosyncratic preferences of a mixed crowd.

1. Introduction

The stochastic multi-armed bandits became popular as a stripped-down model of *exploration versus exploitation* balance in sequential decision problems. In its simplest formulation, we are facing a slot machine with several arms. The rewards of these arms are modeled by unknown but bounded and independent random variables. To maximize our long term reward, we would like to play an arm with maximal expected value but we need to explore efficiently all the arms in order to find it.

The cost of ignorance is traditionally expressed in term of *expected regret* : the expected difference of reward between a playing policy established with perfect knowledge of the environment parameters and a given "unaware" policy. Following Lai & Robbins (1985), several regret analysis have been proposed (see for instance Auer et al., 2002; Audibert et al., 2008; Auer & Ortner, 2011).

Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28.

Another way to evaluate bandit algorithms is to consider *pure-exploration* or PAC *sample complexity*: how to find a nearly-optimal arm with high confidence in a minimum of trials? For multi-armed bandits, several algorithms where already proposed and studied from this perspective in (Even-Dar et al., 2002; 2006).We can also reverse the PAC question to control the prediction accuracy after a fixed number of samples as in (Audibert et al., 2010; Bubeck et al., 2011).

When the number of trials is bounded by a known horizon, we can adopt an *explore then exploit* strategy to control the final regret with an efficient exploration algorithm (Yue et al., 2012). This is the perspective we adopt here.

1.1. Rigged bandits

As an introduction to our *generic exploration* setting we consider the *rigged bandits* problem which is a simplified model for click-fraud in online advertising.

In this variant of multi-armed bandits we know from a reliable source (the barmaid of the casino) that the m best arms of the slot machine have been rigged and only deliver counterfeit money. To maximize our gain, we want to design a sequence of experiments in order to determine and play the $(m+1)^{th}$ best arm as early as possible while avoiding the rigged arms.

The main characteristic of this problem lies in the absence of a direct utility feedback: to estimate our real income we need to know with enough confidence which arms were rigged. This problem also requires what we call a *generic exploration policy*: we do not want to design a new exploration algorithm for each possible fraud-detection criterion we have at hand (see section 5.1 for a formalized example).

1.2. Dueling bandits

The *dueling bandit* problem, introduced by Yue & Joachims (2009) to formalize online learning from preference feedback, shares the *indirect* or *parametric feedback* property with rigged bandits. The initial motivation to depart from the absolute-reward model came from information retrieval evaluation where the implicit feedback by means of click logs is strongly biased by the ranking itself. A solution was proposed by Joachims (2003) to circumvent this problem: by interleaving two ranking models, and checking where the user clicked, one obtains an unbiased – but pairwise – preference feedback. Further experiments were performed in (Chapelle et al., 2012).

The original definition of the dueling bandits problem (Yue et al., 2012; Yue & Joachims, 2011) was built upon strong assumptions about the preference matrix: existence of a *strict linear ordering*, *stochastic transitivity* and *stochastic triangular inequality* (see Yue & Joachims, 2011). An extension of this setting with restricted pairing was proposed by Di Castro et al. (2011), but this extension also assumes the preference matrix to be the byproduct of an inherent value for each arm.

In a situation where the preferences reflects the expression of a mixed crowd, there can be several inconsistencies or *voting paradoxes* which contradict these assumptions. The definition we propose here is more relaxed: we do not assume the existence of a perfect linear order, neither do we assume the existence of an inherent value of arms. We simply try to sample efficiently the preference matrix in order to propose a "best element" similar to the one we would choose with perfect knowledge of the crowd preferences.

Electing a "best element" or a "best linear ordering" from such a preference matrix is a tough but old and well-studied problem (see Charon & Hudry, 2010, for a survey), but the works about online and noisy declinations of this problem are scarce (see however Ravikumar et al., 1987; Feige et al., 1994, for related problems). If we change the election criterion according to the vast social choice theory (see Chevaleyre et al., 2007, for a survey),we can decline the dueling bandits in several unexplored flavors: for instance *Borda bandits, Copeland bandits, Slater bandits, or Kemeny bandits.*

1.3. Toward generic exploration algorithms

The traditional approach to deal with exotic sequential decision problems is to design tailor-made algorithms which handle simultaneously the exploration of the environment and the exploitation of its knowledge. One purpose of this article is to explore the possibility of a generic algorithm which automatically generates an efficient exploration policy for any given decision criterion. We propose a generic algorithm with theoretical guarantees for the case of *parametric decision problems* and we evaluate its performances on relatively simple declinations of rigged and dueling bandits.

2. Main Problem Statement

Consider a stationary environment modeled by a vector of N unknown parameters $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_N) \in [0,1]^N$ (By convention, we write the vectors with bold faces). We have a noisy perception of $\boldsymbol{\mu}$ modeled by a vector $\mathbf{X}$ of N independent random variables $X_i \in [0,1]$ verifying $\mathbb{E}[X_i] = \mu_i$ for each index $i = 1, \ldots, N$. The only hint we may have about $\boldsymbol{\mu}$ is a set of feasible environment configurations $\mathcal{F} \subseteq [0,1]^N$.

Let $\mathcal{D}$ be a set of decisions and $U : \mathcal{D} \times \mathcal{F} \to \mathbb{R}^+$ be a given utility function. From this utility function we can derive a decision function $f : \mathcal{F} \to \mathcal{D}$ which computes an optimal option $f(\mathbf{x}) \in \arg\max_d U(d, \mathbf{x})$ for each feasible realization $\mathbf{x}$ of the random vector $\mathbf{X}$.

The *parametric decision problem* consists in finding the best decision $d^* := f(\boldsymbol{\mu})$ with a high probability after a minimum amount of samples of the environment parameters.

If we have a metric $L : \mathcal{D} \times \mathcal{D} \to \mathbb{R}_+$ to compare decisions, we can also search an ε-*approximation* of d^* (i.e. a decision $d \in \mathcal{D}$ such that $L(d^*, d) \leqslant \varepsilon$).

We use a "budgeted" version of PAC learning:

Definition 1. *An algorithm is an (ε, δ)-PAC algorithm with horizon T for the* parametric decision problem *if it outputs an ε-approximation of d^* with probability at least $1 - \delta$ when it terminates with strictly less than T samples. We call* exploration time *the number of parameter samples required for termination.*

This definition extends PAC learning to finite horizons: to avoid confusion we use the term *exploration time* instead of *sample complexity* when T is finite. The exploration time at horizon T with $\delta = 1/T$ provides an upper bound for the expected cumulative regret (see Appendix B.1 for further details).

The decision function may be the result of quite a complex algorithm, but in the problems we consider here (K-dueling bandits for instance), $\mathcal{D}$ is finite and the decision function is partitioning the input space into single-decision areas. For these problems we will assume that the environment state $\boldsymbol{\mu}$ falls outside of the decision frontiers. In other words, we will assume that

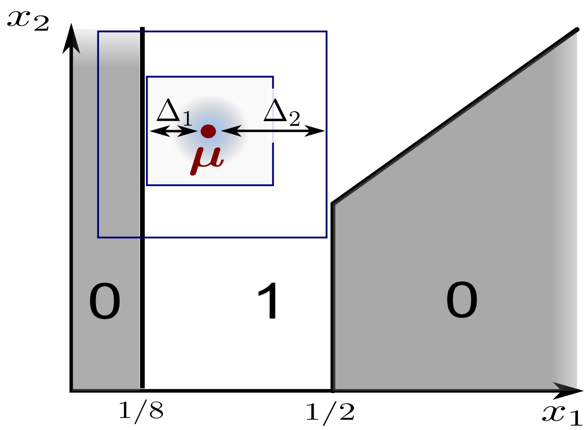

Figure 1. A simple example of binary decision function defined by $f(x_1, x_2) = [x_1 > 1/8 \wedge (x_1 < 1/2 \vee x_2 > x_1)]$ (We use the square brackets $[\cdot]$ to denote the characteristic function of a predicate). How accurately do we need to estimate $\boldsymbol{\mu}$ in order to take the good decision? The environment state $\boldsymbol{\mu}$ is at distance Δ_1 from the nearest decision frontier but we can stop exploring x_2 as soon as we know that $x_1 < 1/2$.

there exists a neighborhood of $\boldsymbol{\mu}$ where f is constant. In order to analyze the performance of our algorithm in the next section, we will need a finer description of this neighborhood. For instance, the binary decision function defined in Figure 1 is constant on the neighborhood of $\boldsymbol{\mu}$ but it is also independent of x_2 for any configuration where $x_1 < 1/2$.

Let us introduce some more notations: hereafter $\|\mathbf{x}\|_\infty := \max_i |x_i|$ will denote the l_∞ norm, $B(\boldsymbol{\mu}, r) := \{\mathbf{x} \mid \|\mathbf{x} - \boldsymbol{\mu}\|_\infty < r\}$ will denote the l_∞ ball (or box) of radius r around $\boldsymbol{\mu}$, and $\mathbf{e_i}$ will denote the standard basis vector with a 1 in the i^{th} coordinate and 0's elsewhere.

Definition 2. *Let $\mathcal{H}$ be a subset of $\mathcal{F}$. The decision function f is* independent of its parameter i on $\mathcal{H}$ *if for any $\alpha \in [-1, +1]$ we have:*

$$\mathbf{x}, \mathbf{x} + \alpha \mathbf{e_i} \in \mathcal{H} \Rightarrow f(\mathbf{x}) = f(\mathbf{x} + \alpha \mathbf{e_i})$$

For instance on Figure 1 the decision is independent of x_2 on the set $B(\boldsymbol{\mu}, \Delta_2)$. This local *independence*, parametrized by the Δ_i radii, captures the sensitivity of the decision to its input parameters around the environment state $\boldsymbol{\mu}$.

3. The SAVAGE Algorithm

We propose a generic zooming algorithm to solve the N dimensional parametric decision problem with high confidence. This algorithm, called **SAVAGE** *(Sensitivity Analysis of VAriables for Generic Exploration)* is described in Algorithm 1. It works by reducing progressively a box-shaped confidence set $\mathcal{H}$ until a single decision remains in $f(\mathcal{H})$. The algorithm stops exploring a parameter when it knows from a sensitivity analysis subroutine **IndepTest**$(f, \mathcal{H}, i)$ that, given our knowledge of the environment, the final decision will not change according to this parameter; in other words when f is independent of i on $\mathcal{H}$ as formalized in Definition 2.

Algorithm 1 SAVAGE algorithm

Input: $\mathbf{X} = (X_1, \ldots, X_N)$, f, $\mathcal{F}$, T, δ
Initialization:
$\mathcal{W} := \{1, \ldots, N\}$, $\mathcal{H} := \mathcal{F}$, $s := 1$
$\forall i \in \mathcal{W}$: $\hat{\mu}_i := 1/2$, and $t_i := 0$
while $\neg$**Accept**$(f, \mathcal{H}, \mathcal{W}) \wedge s \leqslant T$ **do**
  Pick a variable index $i \in \arg\min_{\mathcal{W}} \{t_1, \ldots, t_N\}$
  $t_i := t_i + 1$
  Sample the i^{th} distribution $x_i \leftarrow X_i$
  $\hat{\mu}_i := (1 - \frac{1}{t_i})\hat{\mu}_i + \frac{1}{t_i} x_i$
  $\mathcal{H} := \mathcal{H} \cap \{\mathbf{x} \mid |x_i - \hat{\mu}_i| < c(t_i)\}$
  $\mathcal{W} := \mathcal{W} \backslash \{j \parallel$ **IndepTest**$(f, \mathcal{H}, j)\}$
  $s := s + 1$
end while
return $\hat{d} \in f(\mathcal{H})$

The boundaries of $\mathcal{H}$ are defined by the confidence radius:

$$c(t) = \sqrt{\frac{1}{2t} \log(\frac{\eta(t)}{\delta})}\ , \tag{1}$$

where the η function is set to $2NT$, when the horizon T is finite, and $\frac{\pi^2 N t^2}{3}$ when it is infinite (PAC setting).

Termination is controlled by the predicate:

$$\mathbf{Accept}(f, \mathcal{H}, \mathcal{W}) := \text{"}\mathcal{W} = \varnothing\text{"} \tag{2}$$

$$\text{which implies} \quad |f(\mathcal{H})| = 1 \tag{3}$$

Theorem 1. *If f is independent of each parameter i on $\mathcal{F} \cap B(\boldsymbol{\mu}, \Delta_i)$, with $\Delta_i > 0$, then SAVAGE is a $(0, \delta)$-PAC algorithm with horizon T for the parametric decision problem. When $T = \infty$, its sample complexity is bounded by:*

$$\sum_{i=1}^{N} \mathcal{O}\left(\frac{\log(\frac{N}{\delta \Delta_i})}{\Delta_i^2}\right) .$$

When $T < \infty$, its exploration time is bounded by:

$$\sum_{i=1}^{N} \mathcal{O}\left(\frac{\log(\frac{NT}{\delta})}{\Delta_i^2}\right) .$$

The proof is given in Appendix A.1.

By definition, if f is independent of each parameter i on $B(\boldsymbol{\mu}, \Delta_i)$, then f is constant on the minimal box $B(\boldsymbol{\mu}, \Lambda)$, where $\Lambda = \min_i \Delta_i$. An exploration policy without elimination would reach this neighborhood after $\mathcal{O}\left(\frac{N \log(\frac{NT}{\delta})}{\Lambda^2}\right)$ samples.

This means that SAVAGE will outperform a uniform exploration policy as soon as the Δ_i are not equal. It is also worth noting for practical purpose, that this improvement will hold even if we replace $\mathbf{IndepTest}(f, \mathcal{H}, i)$ with a sufficient condition of independence. Such a relaxation requires however to replace (2) by (3) to ensure termination.

3.1. Independence predicates

The independence predicate $\mathbf{IndepTest}(f, \mathcal{H}, i)$ is a property of the decision function and its feasible set. It can thus be specialized via symbolic calculus or handcrafted for specific problems where the properties of f and $\mathcal{F}$ are well known.

For example in traditional multi-armed bandits settings, when $f(x_1, \ldots, x_N) \in \arg\max\{x_1, \ldots, x_N\}$ and $\mathcal{H}(\mathbf{t})$ is encoded by a product of confidence intervals $[a_i, b_i]$, we can use the SAVAGE algorithm with the following specialized predicate $\mathbf{IndepTest}_f(\mathcal{H}(\mathbf{t}), i)$:

$$(\exists j,\ b_i \leqslant a_j) \vee (\forall k \neq i,\ b_k \leqslant a_i) \ . \tag{4}$$

With this predicate, we fall-back almost to the "arm elimination" of (Even-Dar et al., 2002). We however slightly depart from this algorithm by forcing inclusion of the successive confidence sets: $[a_i, b_i] := [\max\{a_i, \hat{\mu}_i - c(t_i)\}, \min\{b_i, \hat{\mu}_i + c(t_i)\}]$.

If we rather want to retrieve the $(m+1)^{th}$ best arm like in rigged bandits, the independence predicate becomes:

$$\vee \begin{array}{l} (\exists \mathcal{A}, |\mathcal{A}| = m+1,\ \forall j \in \mathcal{A},\ b_i \leqslant a_j) \\ (\exists \mathcal{B}, |\mathcal{B}| = N - m,\ \forall k \in \mathcal{B},\ b_k \leqslant a_i) \ . \end{array} \tag{5}$$

A simple formalization of the independence allows us to apply SAVAGE and Theorem 1 to several other variants of multi-armed bandits.

When the knowledge about f or $\mathcal{F}$ is scarce, and the dimension of the problem is not too high, another solution that we only explored empirically is to estimate the independence predicate by *"introspective"* simulations. We used the multi-start random-walk approximation detailed in Algorithm 2. It provides an "almost-everywhere statement" of the property with an asymmetric risk of failure which can be made arbitrary low by increasing the number of samples (parameters m and M). This kind of method is widely used in sensitivity analysis (see Saltelli et al., 2000, for a survey).

Algorithm 2 Parameters Elimination by Sampling

Input: f,$\mathcal{H}$,$\mathcal{W}$,m,M
Initialization:
$\mathcal{S} \leftarrow \varnothing$
for $l = 1, \ldots, m$ **do**
  Sample $\mathbf{x}$ uniformly from $\mathcal{H}$
  $\mathbf{x}' \leftarrow \mathbf{x}$
  for $s = 1, \ldots, M$ **do**
    Pick a random parameter $i \in \mathcal{W} \backslash \mathcal{S}$
    Re-sample x_i until $\mathbf{x} \in \mathcal{H}$
    if $f(\mathbf{x}) \neq f(\mathbf{x}')$ **then**
      $\mathcal{S} := \mathcal{S} \cup \{i\}$
    end if
    $x'_i \leftarrow x_i$
  end for
end for
$\mathcal{W} := \mathcal{W} \cap \mathcal{S}$

3.2. Approximate decision

If the decision function f is λ-Lipschitz for the decision comparison metric L and the l_∞ norm with a known Lipschitz constant, we are able to relax the problem by searching only an ε-approximation of the best decision. In order to do so we replace the $\mathbf{Accept}(f, \mathcal{H}, \mathcal{W})$ condition by:

$$\forall \mathbf{x}, \mathbf{x}' \in \mathcal{H},\ \|\mathbf{x} - \mathbf{x}'\|_\infty \leqslant \frac{\varepsilon}{\lambda} \tag{6}$$

$$\text{which implies} \qquad \forall d, d' \in f(\mathcal{H}), \quad L\left(d, d'\right) \leqslant \varepsilon$$

For example in multi-armed bandits problem, the decision function is 2-Lipschitz for the utility metric $L(d, d') = |U(d, \boldsymbol{\mu}) - U(d', \boldsymbol{\mu})|$. Indeed, the utility of an arm is its mean reward: $U(i, \boldsymbol{\mu}) = \mu_i$, hence if d^* is the best arm and $i = f(\mathbf{x}) \in \arg\max_j x_j$, then:

$$\begin{aligned} L\left(d^*, f(\mathbf{x})\right) &= |\mu_{d*} - \mu_i| = \mu_{d*} - \mu_i \\ &\leqslant \mu_{d*} - \mu_i + x_i - x_{d*} \\ &\leqslant 2 \cdot \|\boldsymbol{\mu} - \mathbf{x}\|_\infty \ . \end{aligned}$$

Theorem 2. *If f is λ-Lipschitz around the environment state $\boldsymbol{\mu}$, and if f is independent of each parameter i on $\mathcal{F} \cap B(\boldsymbol{\mu}, \Delta_i)$ with radius $\Delta_i > 0$, then SAVAGE with (6) as acceptance condition is an (ε, δ)-PAC algorithm with horizon T for the parametric decision problem. When $T = \infty$, its sample complexity is bounded by:*

$$\sum_{i:\Delta_i \geqslant \varepsilon/\lambda} \mathcal{O}\left(\frac{\log(\frac{N}{\delta \Delta_i})}{\Delta_i^2}\right) + \mathcal{O}\left(\frac{\lambda^2 N_{\varepsilon,\lambda}}{\varepsilon^2} \log(\frac{\lambda N}{\delta \varepsilon})\right);$$

where $N_{\varepsilon,\lambda} = |\{i \mid \Delta_i < \varepsilon/\lambda\}|$.
When $T < \infty$, *its exploration time is bounded by:*

$$\sum_{i:\Delta_i > \varepsilon/\lambda} \mathcal{O}\left(\frac{\log(\frac{NT}{\delta})}{\Delta_i^2}\right) + \mathcal{O}\left(\frac{\lambda^2 N_{\varepsilon,\lambda}}{\varepsilon^2}\log(\frac{NT}{\delta})\right) \ .$$

See Appendix A.2 for the proof. The access to the decision metric L may also be explicit in which case we can use directly $\forall d, d' \in f(\mathcal{H}),\quad L(d,d') \leqslant \varepsilon$, in place for (6) but it is difficult to dispense with the Lipschitz hypothesis to obtain a generic bound.

4. Application to K-armed Dueling Bandits

From now on, we call $K{\times}K$ *preference matrix* a K×K matrix $(x_{i,j})$ such that $x_{i,j} + x_{j,i} = 1$ for each $i,j \in \{1,\dots,K\}$ (we use lower-case letters to match the notations of Section 2).

The K-dueling problem, as presented in (Yue et al., 2012; Yue & Joachims, 2011), assumes the existence of an environment preference matrix $\boldsymbol{\mu}$ from which we only have a noisy perception modeled, as in (Feige et al., 1994), by a K×K-matrix of random variables $X_{i,j} \in [0,1]$ verifying $\mathbb{E}[X_{i,j}] = \mu_{i,j}$. Our aim is to design a sequence of pairwise experiments (i_t, j_t) called *duels* for $t = 1,\dots,T$ in order to find the best arm. They also assume the following properties for the preference matrix(WLOG for a proper indexation of the matrix):

strict linear order: if $i < j$ then $\mu_{i,j} > \frac{1}{2}$;

γ-relaxed stochastic transitivity: if $1 < j < k$ then $\gamma \cdot \mu_{1,k} \geqslant \max\{\mu_{1,j}, \mu_{j,k}\}$;

stochastic triangular inequality: if $1 < j < k$ then $\mu_{1,k} \leqslant \mu_{1,j} + \mu_{j,k} - \frac{1}{2}$.

These last three assumptions are realistic when the preference matrix is the result of a perturbed linear order. This is indeed the case for some generative models where the number of parameters of the environment is assumed to be K: the inherent values of arms. In a situation where the preferences may contain cycles (or *voting paradoxes*) there is no clear notion of what the best arm is, and the notion of regret is unclear.

To avoid these problems, we propose to consider a *"voting"* variant of K-dueling bandits where a pairwise election criterion is used to determine the best candidate from the preference matrix. Several election systems can be used, but we will focus here on a simple and well-established one: the Copeland pairwise aggregation method (see Charon & Hudry, 2010).

4.1. Copeland bandits

If $\mathbf{x}$ is a K×K preference matrix, we define the *Copeland score* of an arm i by its number of one-to-one majority victories:

$$U_{Cop}(i,\mathbf{x}) = \sum_j [x_{i,j} > \frac{1}{2}] \ . \qquad (7)$$

Any element of $\arg\max_i U_{Cop}(i,\mathbf{x})$ is called a *Copeland winner* of the matrix.

4.1.1. General Copeland bandits

With a preference matrix of size K we have $N = K(K-1)/2$ free parameters to estimate: we can encode $\mathcal{H}$ as a product of intervals $[a_{i,j}, b_{i,j}]$ and apply the SAVAGE algorithm with $\mathbf{IndepTest}_f(\mathcal{H},(i,j)) =$

$$\begin{aligned} &(a_{i,j} > \tfrac{1}{2} \vee b_{i,j} < \tfrac{1}{2}) \vee \mathbf{Cop}(\mathcal{H},(i,j)) \ , \\ &\text{where } \mathbf{Cop}(\mathcal{H},(i,j)) := \exists i^+ \text{ s. t.}, \\ &\quad (\min U_{cop}(i^+,\mathcal{H}) > \max U_{cop}(i,\mathcal{H})) \\ &\wedge \ (\min U_{cop}(i^+,\mathcal{H}) > \max U_{cop}(j,\mathcal{H})) \ . \end{aligned} \qquad (8)$$

By applying Theorem 1, we obtain an exploration time bound of order:

$$\sum_{i<j} \mathcal{O}\left(\frac{\log(KT/\delta)}{\Delta_{i,j}^2}\right) \leqslant \mathcal{O}\left(K^2 \frac{\log(KT/\delta)}{\Lambda^2}\right) \ , \qquad (9)$$

where $\Delta_{i,j} = |\mu_{i,j} - \frac{1}{2}|$ for any $i < j$, and $\Lambda = \min_{i<j} \Delta_{i,j}$. This bound requires weak assumptions about the preference matrix $\boldsymbol{\mu}$ but its strong dependence on the "hard" parameters (when $\mu_{i,j}$ is close to $\frac{1}{2}$) makes it quite conservative. The behavior of the algorithm is more efficient in practice.

4.1.2. Condorcet assumption

If there exists an arm $f(\mathbf{x})$ preferred to all the others, it is unique and verifies $U_{Cop}(f(\mathbf{x}),\mathbf{x}) = K - 1$. The existence of this arm, called *Condorcet winner* of the matrix, allows us to tighten the exploration bound.

Property 1. *If the environment state* $\boldsymbol{\mu}$ *admits arm* i^* *as a Condorcet winner with* $\Delta = \min_{j\neq i^*} \mu_{i^*,j} - \frac{1}{2}$ *and* $\Delta_{i,j} = \max\{\Delta, |\mu_{i,j} - \frac{1}{2}|\}$ *then* f *is independent of* $x_{i,j}$ *on* $B(\boldsymbol{\mu}, \Delta_{i,j})$ *for any* $i < j$.

See Appendix A.3 for the proof.

By applying Theorem 1 when Property 1 holds, we obtain a bound which is less sensitive to the presence of tight duels than (9) without changing the algorithm.

If we know the existence of a Condorcet winner, we can also tame the SAVAGE algorithm by restricting the feasible set $\mathcal{F}$ to the K×K-preferences matrices admitting a *Condorcet winner*:

$$\mathcal{F}_{Cond} := \{\mathbf{x} \mid \exists i^*, U_{Cop}(i^*, \mathbf{x}) = K - 1\} \ . \tag{10}$$

We can obtain a formal independence test in $\mathcal{F}_{\text{cond}}$ by replacing (8) with:

$$\begin{array}{ll} & (\max_{\mathbf{x}\in\mathcal{H}} U_{Cop}(i, \mathbf{x}) < K - 1) \\ \vee & (\max_{\mathbf{x}\in\mathcal{H}} U_{Cop}(j, \mathbf{x}) < K - 1) \end{array} \tag{11}$$

To stop exploration with an ε-approximation[1] of the winner, we replace $\mathbf{Accept}(f, \mathcal{H}, \mathcal{W})$ by:

$$\forall \mathbf{x} \in \mathcal{H}, \ K - 1 - U_{Cop}(f(\mathbf{x}), \mathbf{x}) \leqslant \varepsilon \ . \tag{12}$$

Theorem 3. *If the environment state $\boldsymbol{\mu}$ is known to admit a Condorcet winner $i^* = f(\boldsymbol{\mu})$ then SAVAGE with $\mathcal{F}_{Cond}$ as feasible set and (12) as acceptance condition is an (ε, δ)-PAC algorithm with horizon T for the Copeland bandits problem. When $T = \infty$, its samples complexity is bounded by:*

$$\sum_{j=\varepsilon+1}^{K-1} \mathcal{O}\left(\frac{j \cdot \log\left(\frac{K}{\delta\Delta_j}\right)}{\Delta_j^2}\right) \ .$$

Where for each $j \neq i^*$ we have $\Delta_j = \mu_{i^*,j} - \frac{1}{2}$ (indexed WLOG by increasing values of Δ_j).
When $T < \infty$ its exploration time is bounded by:

$$\sum_{j=\varepsilon+1}^{K-1} \mathcal{O}\left(\frac{j \cdot \log(\frac{KT}{\delta})}{\Delta_j^2}\right) \ .$$

A proof is given in Appendix A.4 This is a significant improvement from (9) but it does not remove the quadratic term K^2. This leading K^2 factor is the price we pay for accepting less constrained preference matrices.

4.2. Borda bandits

Another simple way to elect the winner of the matrix is to use Borda count. Each competitor is ranked according to its mean performance against others:

$$U_{Bor}(i, \mathbf{x}) = x_{i,\cdot} = \sum_j x_{i,j} \ . \tag{13}$$

The main advantage of this criterion is that it both offers stability (the utility is linear) and clearly reduces the dimension of the problem to only K parameters: $x_{i,\cdot}$ for $i = 1, \ldots, K$. This means that we can simply wrap a classical bandit algorithm to search for the Borda winner of the matrix. It is quite easy however to design a Condorcet preference matrix where the Borda winner is not the Condorcet winner[2].

[1] ε is the number of tolerated defeats.

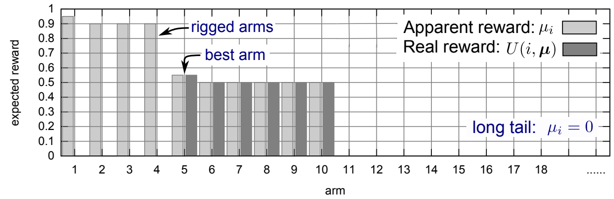

Figure 2. A 100-armed rigged bandit designed in order to game simple exploration algorithms : the four apparently best arms only deliver counterfeit money. The algorithm must sample intensively the arms 5 to 10 in order to guess that 5 is the best.

The decision criterion underlying the *Beat the Mean Bandit* algorithm proposed by (Yue & Joachims, 2011) is different: the matrix rows are explored to find Borda loosers which are progressively eliminated from the matrix until only one arm remains. This election procedure called *Bottom-up Borda elimination* returns the Condorcet winner if there exists one. SAVAGE being a generic algorithm, it can be applied directly to these two voting criteria.

5. Simulations

Algorithm 3 Online sampling evaluation process

```
for t = 1, ..., T do
    Choose a parameter i and explore X_i outcome
    Choose decision d̂_t accordingly
    Get unknown reward U(d̂_t, μ)
end for
```

In order to compare algorithms of different natures, we used the pure-exploration online setting described in Algorithm 3. We considered both the best-decision rate and the regret $U(d^*, \boldsymbol{\mu}) - U(\hat{d}_t, \boldsymbol{\mu})$. For all algorithms except *Interleave Filtering* and *Beat the Mean*, we took $\hat{d}_t := f(\hat{\boldsymbol{\mu}}(t))$. For PAC algorithms, we took $\epsilon = 0$ and $\delta = 1/T$ (explore-then-exploit setting). The sample time where the best-decision rate reaches $1 - \delta$ gives an empirical estimation of the PAC exploration time. To avoid nasty side-effects, we shuffled the matrices/parameters at each run.

[2] There exist also K-armed stochastic bandit settings which are non transitive (Gardner, 1970).

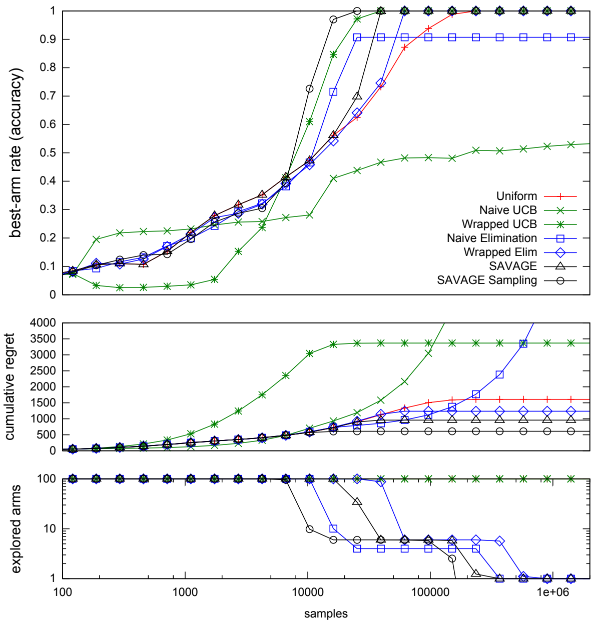

Figure 3. Behavior of the different algorithms for 1000 simulations with Figure 2 distribution. The top figure depicts the best-arm rate, the middle figure show the cumulative regret and the bottom one tracks the number of active arms for elimination algorithms. Time scale is logarithmic.

5.1. Bandits simulations

For bandits problems the decision space and the exploration space coincide, but we are here in a pure exploration setting where the arm we predict to be the best is not necessary the one we explore. We considered the following algorithms for our bandits simulations:

Uniform: baseline uniform exploration policy (each arm is explored once in a round-robin manner);

Naive UCB: UCB1 (as in Auer et al., 2002);

Naive Elimination: applies Action elimination algorithm (as described in Even-Dar et al., 2002);

Wrapped UCB: applies UCB1 to the wrapped reward random variable $\hat{U}_i = U(i, \hat{\boldsymbol{\mu}})$ used as a proxy for $U(i, \boldsymbol{\mu})$;

Wrapped Elimination: applies Action elimination with the above "wrapped reward";

SAVAGE: applies Algorithm 1 with $\eta(t) := 2NT$ and predicate (5) with $m = 4$;

SAVAGE Sampling: Algorithm 1 with a sampled independence predicate and 1000 simulations by arm (see Algorithm 2).

We compared these algorithms on several Bernoulli reward distributions. We give here the simulation result for a rigged bandits problem specially designed in order to illustrate the different exploration behaviors of the algorithms. In this setting, the utility of arm i (indexed WLOG by decreasing μ_i) is defined by $U(i, \boldsymbol{\mu}) = 0$ if $i < 5$ and $U(i, \boldsymbol{\mu}) = \mu_i$ otherwise (see Figure 2 for the reward distribution, and Figure 3 for the simulations results). As expected, the maximizing policies like *UCB* explore aggressively the head of the distribution but after around 10^3 samples fall into the rigged arms and neglect the second part of the distribution head. The wrapped versions are less sensitive to this trap, but the non-linearity of the utility and the violation of independence it induces both cripple their regret performances in the beginning of the runs. As expected, the **SAVAGE** versions perform well, more surprising is the side-effect of sampling which makes the algorithm more aggressive against weak arms.

5.2. Dueling bandits simulations

For the dueling bandits simulations, we considered **Uniform**, **SAVAGE**, and **SAVAGE Sampling** policies plus the following ones:

SAVAGE/Condorcet: Algorithm 1 with (11) for the independence test;

SAVAGE Sampling/Borda: Algorithm 1 with sampled oracle and a Borda relaxation of the Condorcet feasible set, i.e. $\exists i, \sum_j x_{i,j} > K/2$;

Interleave Filtering: as in (Yue et al., 2012, Alg. 2) with $\hat{b}$ for $\hat{d}_t$;

Beat the Mean: as in (Yue & Joachims, 2011) with $\arg\max\{\hat{P}_b \mid b \in \mathcal{W}_l\}$ for $\hat{d}_t$.

5.2.1. Condorcet simulations

We first used *"hard"* $K \times K$ Condorcet preference matrices $\boldsymbol{\mu}$ defined by $\mu_{i,j} = \frac{1}{2} + j/(2K)$ for each $i < j$. The matrices of this family verify all the assumptions defined in Section 4 but they also offer some difficult duels involving the Condorcet winner (for instance if $K = 100$ we have $\mu_{1,2} = 0.51$ hence $\Delta = 0.01$).

The results of these experiments appear in Figure 4 and Figure 5. The SAVAGE Sampling policy slightly improves from the formal version but its heavy introspection cost makes it difficult to deploy on high-dimension problems. Low-dimension instance of Figure 4 is not favorable for *Interleave Filtering* and *Beat the Mean* which were designed to drop the $\mathcal{O}(K^2)$ term

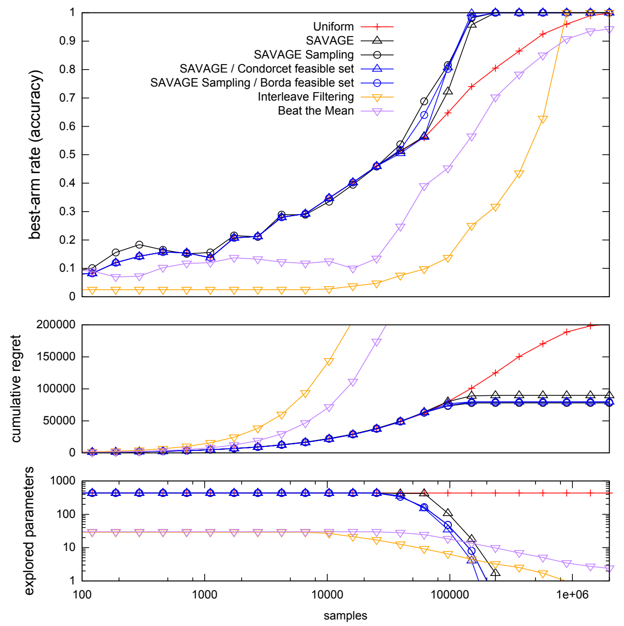

Figure 4. Average good prediction rate and regret for 500 simulations of a 30-armed Condorcet bandit instance. We used the Copeland index (7) to compute the regret.

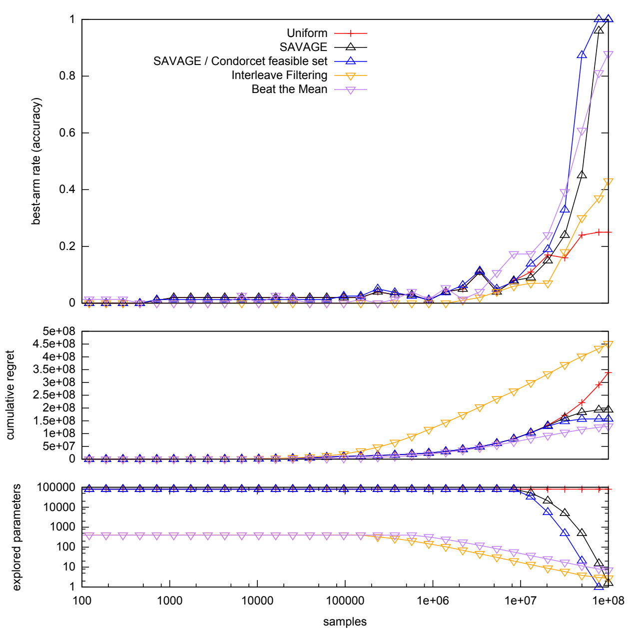

Figure 5. Same setting as in Figure 4 with $K = 400$ and the horizon set to 10^8. When we increase K the problem becomes difficult.

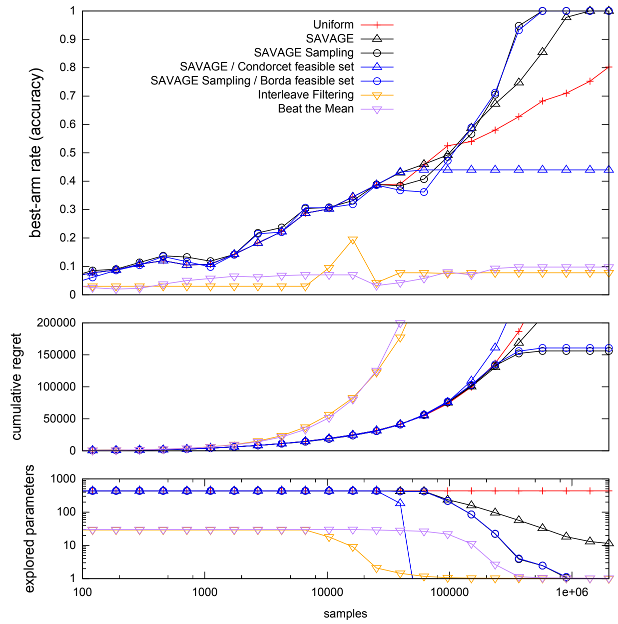

Figure 6. Result of 500 simulations with 30×30 randomized preference matrices.

of the bound with partial – hence risky – exploration strategies (see Yue & Joachims, 2011).

5.2.2. General case simulations

In order to study the behavior of the algorithms with more realistic – non-Condorcet – preferences, we generated uniformly random preference matrices and performed the same experiments. As expected, when the Condorcet hypothesis is violated the performances of specialized algorithms (including SAVAGE/ Condorcet) collapse well behind the baseline uniform exploration policy (see Figure 6).

6. Conclusion

We proposed **SAVAGE**, a flexible and generic algorithm based on sensitivity analysis of parameters for online learning with indirect feedback. We provided PAC theoretical guarantees for this algorithm when used with proper independence predicates. We also proposed a generic "introspective sampling" method to approximate theses predicates.

Our simulations confirmed and reinforced the theoretical results on various parametric decision problems from classical bandits to K-armed dueling bandits.

The "voting bandits" framework we proposed naturally extends dueling bandits for realistic situations where the preferences reflects mixed and inconsistent opinions. The SAVAGE algorithm is robust and clearly outperforms state-of-the art algorithms in such situations.

The construction of a generic exploration algorithm reaching optimality for any provided decision function remains as a challenging open problem.

Acknowledgments

We would like to thank the reviewers for their careful reading and helpful comments.

References

Audibert, J.Y., Munos, R., and Szepesvári, Cs. Exploration-exploitation trade-off using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 2008.

Audibert, J.Y., Bubeck, S., and Munos, R. Best arm identification in multi-armed bandits. In *COLT*, Haïfa (Israël), 2010. Omnipress.

Auer, P. and Ortner, R. UCB revisited: Improved regret bounds for the stochastic multi-armed bandit problem. *Period.Math.Hungar.*, 61(1–2):55–65, 2011.

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

Bubeck, S., Munos, R., and Stoltz, G. Pure exploration in finitely-armed and continuous-armed bandits. *Theor. Comput. Sci.*, 412(19):1832–1852, 2011.

Chapelle, O., Joachims, T., Radlinski, F., and Yue, Y. Large-scale validation and analysis of interleaved search evaluation. *ACM Trans. Inf. Syst.*, 30(1):6, 2012.

Charon, I. and Hudry, O. An updated survey on the linear ordering problem for weighted or unweighted tournaments. *Annals OR*, 175(1):107–158, 2010.

Chevaleyre, Y., Endriss, U., Lang, J., and Maudet, N. A short introduction to computational social choice. In *SOFSEM*, volume 4362 of *LNCS*, pp. 51–69. Springer-Verlag, 2007.

Di Castro, D., Gentile, C., and Mannor, S. Bandits with an edge. *CoRR*, abs/1109.2296, 2011.

Even-Dar, E., Mannor, S., and Mansour, Y. PAC bounds for multi-armed bandit and Markov decision processes. In *COLT*, pp. 255–270, London, UK, 2002. Springer-Verlag.

Even-Dar, E., Mannor, S., and Mansour, Y. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *JMLR*, 7:1079–1105, 2006.

Feige, U., Raghavan, P., Peleg, D., and Upfal, E. Computing with noisy information. *SIAM J. Comput.*, 23(5):1001–1018, 1994.

Gardner, M. Mathematical games: The paradox of the nontransitive dice and the elusive principle of indifference. *Scientific American*, 223:110–114, dec 1970.

Hoeffding, W. Probability inequalities for sums of bounded random variables. *J. of the American Statistical Association*, 58(301):13–30, 1963.

Joachims, T. Evaluating retrieval performance using clickthrough data. In *Text Mining*, pp. 79–96. Physica/Springer Verlag, 2003.

Lai, T.L. and Robbins, H. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985.

Ravikumar, B., Ganesan, K., and Lakshmanan, K.B. On selecting the largest element in spite of erroneous information. In *STACS*, volume 247 of *LNCS*, pp. 88–99. Springer Berlin Heidelberg, 1987.

Saltelli, A., Chan, K., and Scott, E.M. (eds.). *Sensitivity analysis*. Wiley series in probability and statistics. J. Wiley & sons, New York, Chichester, Weinheim, 2000.

Yue, Y. and Joachims, T. Interactively optimizing information retrieval systems as a dueling bandits problem. In *ICML*, volume 382 of *ACM Proceeding Series*, pp. 1201–1208. ACM, 2009.

Yue, Y. and Joachims, T. Beat the mean bandit. In *ICML*, pp. 241–248. Omnipress, 2011.

Yue, Y., Broder, J., Kleinberg, R., and Joachims, T. The k-armed dueling bandits problem. *J. Comput. Syst. Sci.*, 78(5):1538–1556, 2012.

A. Proofs

A.1. Proof of Theorem 1

Theorem 1.
1 - Correctness
We first establish that the unknown parameter value $\boldsymbol{\mu}$ will stay inside the confidence set during all the computation with a probability of at least $1-\delta$. Let $\mathbf{t} = (t_1, \ldots, t_N)$ be a configuration of the algorithm. We denote by $\mathcal{H}(\mathbf{t})$ its corresponding hypothesis set:

$$\mathcal{H}(\mathbf{t}) = \mathcal{F} \cap \bigtimes_i \bigcap_{n<t_i} \{\mathbf{x} \mid \quad |x_i - \hat{\mu}_i(n)| < c(t_i)\} \tag{14}$$

where the $\hat{\mu}_i(n)$ is used to denote the ML estimate of μ_i after n samples. We can bound the probability of failure:

$$\mathbb{P}[\exists \mathbf{t},\ \boldsymbol{\mu} \notin \mathcal{H}(\mathbf{t})] \leqslant \mathbb{P}\left[\exists i, \exists t < T \quad \text{s.t. } |\mu_i - \hat{\mu}_i(t)| > c(t)\right] \ ,$$

and by the union bound:

$$\mathbb{P}[\exists \mathbf{t},\ \boldsymbol{\mu} \notin \mathcal{H}(\mathbf{t})] \leqslant \sum_{i=1}^{N} \sum_{1 \leqslant t \leqslant T} \mathbb{P}[|\mu_i - \hat{\mu}_i(t)| > c(t)] \ . \tag{15}$$

By applying Höeffding Lemma (Hoeffding, 1963) and equation (1), we obtain:

$$\mathbb{P}[|\mu_i - \hat{\mu}_i(t)| > c(t)] \leqslant 2e^{-2tc^2(t)} \leqslant \frac{2\delta}{\eta(t)} \ .$$

Therefore, $\eta(t)$ being properly chosen according to (1), we have:

$$\mathbb{P}[\exists \mathbf{t},\ \boldsymbol{\mu} \notin \mathcal{H}(\mathbf{t})] \leqslant \sum_{i=1}^{N} \sum_{1 \leqslant t \leqslant T} \frac{2\delta}{\eta(t)} = \delta \ .$$

In particular, when $T = \infty$, we have $\sum_{t \geqslant 1} \frac{1}{t^2} = \frac{\pi^2}{6}$.

From now on we will assume that $\boldsymbol{\mu} \in \mathcal{H}$ at any step of the algorithm.

2 - Sample complexity / exploration time
We are left to compute the sample complexity. We first treat the finite horizon case where $\eta(t) = 2NT$. Let us assume that parameter i was not eliminated after $\frac{2}{\Delta_i^2} \log\left(\frac{2NT}{\delta}\right)$ sampling rounds, then:

$$\begin{aligned} & t_i > \frac{2}{\Delta_i^2} \log\left(\frac{2NT}{\delta}\right) \Leftrightarrow \frac{1}{2t_i} \log\left(\frac{2NT}{\delta}\right) < \frac{\Delta_i^2}{4} \\ & \Rightarrow c(t_i) < \frac{\Delta_i}{2} \ . \end{aligned}$$

Let $\mathbf{x}$ be a point in $\mathcal{H}$, and let α be a real such that $\mathbf{x} + \alpha \mathbf{e_i} \in \mathcal{H}$. The point $\mathbf{x}$ can be projected into a new point:

$$\mathbf{x}' := \mathbf{x} + \sum_{j \notin \mathcal{W}} (\mu_j - x_j)\mathbf{e_j} \tag{16}$$

Because $\mathcal{H}$ is the intersection of all the previous hypothesis sets, f is independent of any parameters $j \notin \mathcal{W}$ on $\mathcal{H}$, hence $f(\mathbf{x}') = f(\mathbf{x})$.

By construction of the algorithm (a *round-robin* allocation), when the oracle is called, for all $j \in \mathcal{W}$ we have $t_i \leqslant t_j$. The function $c(t)$ being strictly decreasing, we also have:

$$[\hat{\mu}_j(t_j) - c(t_j), \hat{\mu}_j(t_j) + c(t_j)] \subseteq [\mu_j - \Delta_i, \mu_j + \Delta_i] \tag{17}$$

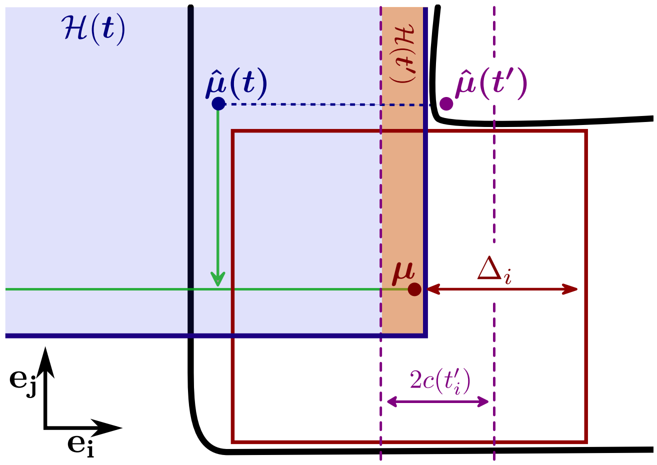

Figure 7. The SAVAGE algorithm stops exploring a parameter j at configuration time $\mathbf{t}$, when the decision function is independent of x_j in the confidence set $\mathcal{H}(\mathbf{t})$. The point estimate $\hat{\boldsymbol{\mu}}(\mathbf{t})$ is in the middle of $\mathcal{H}(\mathbf{t})$. The independence property is illustrated by the bold frontier which crosses vertically $\mathcal{H}(\mathbf{t})$ on the left of the figure. Later, at configuration time $\mathbf{t}'$, when $2c(t'_i) < \Delta_i$, the j^{th} coordinate of the new estimation point $\hat{\boldsymbol{\mu}}(\mathbf{t}')$ is unchanged. This estimation point may be outside of the previous confidence set $\mathcal{H}(\mathbf{t})$ and violate the independence w.r.t. j, but any point in the intersection set $\mathcal{H}(\mathbf{t}') \subseteq \mathcal{H}(\mathbf{t})$ obtains the same decision as its projection to the affine hyperplane perpendicular to $\mathbf{e_j}$ which contains $\boldsymbol{\mu}$.

and therefore $\mathbf{x}' \in B(\boldsymbol{\mu}, \Delta_i)$. By hypothesis $f(\mathbf{x}') = f(\mathbf{x}' + \alpha\mathbf{e_i})$, hence $f(\mathbf{x} + \alpha\mathbf{e_i}) = f(\mathbf{x})$: f is independent of i on $\mathcal{H}$ and will be eliminated at this round. This argument is illustrated by Figure 7.

In the worst case, all parameter will be eliminated after at most $\sum_{i=1}^{N} \frac{2}{\Delta_i^2} \log\left(\frac{2NT}{\delta}\right)$ samples.

When T is infinite, we assume that parameter i was not eliminated after $\frac{16}{\Delta_i^2} \log\left(\frac{\pi^2 N}{3\delta\Delta_i}\right)$ sampling steps which ensures that $c(t_i) < \Delta_i/2$.

Indeed, starting from:

$$t_i \geqslant \frac{16}{\Delta_i^2} \log\left(\frac{\pi^2 N}{3\delta\Delta_i}\right) \ , \tag{18}$$

If we apply the decreasing function

$$c^2(t) = \frac{1}{2t} \log\left(\frac{\pi^2 N t^2}{3\delta}\right)$$

to both sides of (18) inequality, we obtain:

$$c^2(t_i) \leqslant \frac{\Delta_i^2}{32 \log\left(\frac{\pi^2 N}{3\delta\Delta_i}\right)} \cdot A$$

where:

$$\begin{aligned}
A &= \log\left(\frac{\pi^2 N}{3\delta}\right) + 2\log\left(\frac{16}{\Delta_i^2}\log\left(\frac{\pi^2 N}{3\delta\Delta_i}\right)\right) \\
A &= \log\left(\frac{\pi^2 N}{3\delta}\right) + 4\log\left(\frac{4}{\Delta_i}\right) + 2\log\log\left(\frac{\pi^2 N}{3\delta\Delta_i}\right) \\
A &\leqslant 4\log\left(\frac{\pi^2 N}{3\delta\Delta_i}\right) + 8\log(2) + 2\log\log\left(\frac{\pi^2 N}{3\delta\Delta_i}\right)
\end{aligned}$$

For any $x \geqslant 6$ we have

$$8 \log(2) + 2 \log\log(x) < 4 \log(x) \ ,$$

hence for any $N \geqslant 2$, $0 \leqslant \delta \leqslant 1$, and $0 \leqslant \Delta_i \leqslant 1$:

$$8 \log(2) + 2 \log\log\left(\frac{\pi^2 N}{3\delta\Delta_i}\right) \leqslant 4 \log\left(\frac{\pi^2 N}{3\delta\Delta_i}\right)$$

and finally

$$c^2(t_i) \leqslant \frac{\Delta_i^2}{32 \log\left(\frac{\pi^2 N}{3\delta\Delta_i}\right)} \left[8 \log\left(\frac{\pi^2 N}{3\delta\Delta_i}\right)\right]$$

$$c^2(t_i) \leqslant \frac{\Delta_i^2}{4}$$

In a worst-case scenario, all parameters will be eliminated after at most $\sum_{i=1}^{N} \frac{16}{\Delta_i^2} \log\left(\frac{\pi^2 N}{3\delta\Delta_i}\right)$ samples. □

A.2. Proof for Theorem 2

Theorem 2. This proof is very similar to the one of Theorem 1. For the correctness we use exactly the same arguments as in Appendix A.1.

For the exploration time we must consider two cases: either $\Delta_i \geqslant \varepsilon/\lambda$ or $\Delta_i < \varepsilon/\lambda$. The former case is similar to Theorem 1.

In the latter case, we replace Δ_i by ε/λ to obtain the individual parameters sampling bounds. Indeed, when T is finite we need at least $t_i \geqslant \frac{2\lambda}{\varepsilon^2} \log\left(\frac{2NT}{\delta}\right)$ samples of parameter i to guarantee $c(t_i) < \frac{\varepsilon}{2\lambda}$.

When $T = \infty$, we obtain the same result after

$$t_i \geqslant \frac{16\lambda^2}{\varepsilon_i^2} \log\left(\frac{\pi^2 N\lambda}{3\delta\varepsilon}\right)$$

samples of parameter i.

We then use the same argument as in (16) to project the problem into a box-shaped neighborhood $B(\hat{\mu}, \frac{\varepsilon}{2\lambda}) \subseteq B(\boldsymbol{\mu}, \frac{\varepsilon}{\lambda})$ where

$$L\left(f(\mathbf{x}), f(\mathbf{x}')\right) \leqslant \lambda \cdot \|\mathbf{x} - \mathbf{x}'\|_\infty \leqslant \varepsilon \ .$$

As a result, the approximate termination condition $\mathcal{H} \subseteq B(\boldsymbol{\mu}, \frac{\varepsilon}{\lambda})$ is guaranteed after at most

$$\sum_{i:\Delta_i \geqslant \varepsilon/\lambda} \frac{2}{\Delta_i^2} \log\left(\frac{2NT}{\delta}\right) + \sum_{i:\Delta_i < \varepsilon/\lambda} \frac{2\lambda^2}{\varepsilon^2} \log\left(\frac{2NT}{\delta}\right) =$$
$$\sum_{i:\Delta_i \geqslant \varepsilon/\lambda} \frac{2}{\Delta_i^2} \log\left(\frac{2NT}{\delta}\right) + \frac{2\lambda^2 N_{\varepsilon,\lambda}}{\varepsilon^2} \log\left(\frac{2NT}{\delta}\right)$$

samples when T is finite.

And after

$$\sum_{i:\Delta_i \geqslant \varepsilon/\lambda} \frac{16}{\Delta_i^2} \log\left(\frac{\pi^2 N}{3\delta\Delta_i}\right) + \sum_{i:\Delta_i < \varepsilon/\lambda} \frac{16\lambda^2}{\varepsilon_i^2} \log\left(\frac{\pi^2 N\lambda}{3\delta\varepsilon}\right) =$$
$$\sum_{i:\Delta_i \geqslant \varepsilon/\lambda} \frac{16}{\Delta_i^2} \log\left(\frac{\pi^2 N}{3\delta\Delta_i}\right) + \frac{16\lambda^2 N_{\varepsilon,\lambda}}{\varepsilon^2} \log\left(\frac{\pi^2 N\lambda}{3\delta\varepsilon}\right)$$

samples when $T = \infty$. □

A.3. Proof of Property 1

Property 1. Two possibilities:

(a) $\Delta_{i,j} = |\mu_{i,j} - \frac{1}{2}|$, or

(b) $\Delta_{i,j} = \Delta = \min_{j \neq i*} \mu_{i*,j} - \frac{1}{2}$.

In the former case for any $x_{i,j} \in [\mu_{i,j} - \Delta_{i,j}, \mu_{i,j} + \Delta_{i,j}]$, we have $\left[x_{i,j} > \frac{1}{2}\right] = \left[\mu_{i,j} > \frac{1}{2}\right]$ hence the independence w.r.t. $x_{i,j}$ on $B(\boldsymbol{\mu}, \Delta_{i,j})$.

In the latter case, for any $j \neq i^*$, we have $B(\boldsymbol{\mu}, \Delta_{i,j}) \subseteq B(\boldsymbol{\mu}, \mu_{i*,j} - \frac{1}{2})$ hence $f\left(B(\boldsymbol{\mu}, \Delta)\right) = \{i^*\}$: we have the independence w.r.t. any parameter on $B(\boldsymbol{\mu}, \Delta)$. □

A.4. Proof of Theorem 3

Theorem 3. 1 - Domino arms elimination
Suppose WLOG that $i^* = f(\boldsymbol{\mu}) = 1$. We will show that if the parameter $x_{1,l}$ is eliminated then all parameters of the form $x_{i,l}$ or $x_{l,j}$ will be automatically eliminated at the same round.

Let s_l be the number of samples required to obtain $c(s_l) \leqslant \Delta_l/2$, hence

$$[\hat{\mu}_{1,l} - c(s_l), \hat{\mu}_{1,l} + c(s_l)] \subseteq [\mu_{1,l} - \Delta_l, \mu_{1,l} + \Delta_l]$$

and therefore ensure with high confidence that l is a loosing arm. In other words if $t_{1,l} > s_l$ then $\forall \mathbf{x} \in \mathcal{H}(\mathbf{t})$ we have: $x_{1,l} > \frac{1}{2}$. Consider the following two facts:

fact 1: $\forall \mathbf{x} \in \mathcal{H}(\mathbf{t}), f(\mathbf{x}) \neq l$ (because $x_{l,1} \leqslant \frac{1}{2}$);

fact 2: $\forall \mathbf{x} \in \mathcal{H}(\mathbf{t}), \forall j \neq f(\mathbf{x})$: $\mathbf{x}_{f(\mathbf{x}),j} > \frac{1}{2}$ (by def. of $\mathcal{F}_{Cond}$ and f).

Let $\mathbf{x} \in \mathcal{H}(\mathbf{t})$ and $\mathbf{x}' \in \mathcal{H}(\mathbf{t})$ a "perturbation" of $\mathbf{x}$ along the $x_{\cdot,l}$ and $x_{l,\cdot}$ parameters:

$$\forall i, j \quad i \neq l \wedge j \neq l \Rightarrow x'_{i,j} = x_{i,j} \ . \tag{19}$$

Suppose that $f(\mathbf{x}) \neq f(\mathbf{x}')$. From *Fact 1* we have both $f(\mathbf{x}) \neq l$ and $f(\mathbf{x}') \neq l$ hence by applying (19) and the "symmetry" of preference matrices, we have:

$$x'_{f(\mathbf{x}'),f(\mathbf{x})} = 1 - x_{f(\mathbf{x}),f(\mathbf{x}')} \tag{20}$$

According to *Fact 2*, we have both $x'_{f(\mathbf{x}'),f(\mathbf{x})} > \frac{1}{2}$ and $x_{f(\mathbf{x}),f(\mathbf{x}')} > \frac{1}{2}$ which contradicts (20). As a consequence, the **IndepTest** predicate will be true for all parameters involving the l^{th} arm.

2 - Summation
Let s_j be such that $t \geqslant s_j \Rightarrow c(t) \leqslant \Delta_j/2$. For instance if T is finite we take $s_j := \frac{2}{\Delta_j} \log\left(\frac{2NT}{\delta}\right)$ (see Appendix A.1).

If the Δ_j radii are indexed by increasing values, the first column elimination will append at least after $s_{K-1} = \frac{2}{\Delta_j} \log\left(\frac{2NT}{\delta}\right)$ full-matrix sampling rounds. This will cost a total of $\frac{K(K-1)}{2} s_{K-1}$ samples. In a worst case, the second elimination will cost $\frac{(K-1)(K-2)}{2}(s_{K-2} - s_{K-1})$ samples, and so on with a cost of order $\frac{j(j+1)}{2}(s_j - s_{j+1})$ at each elimination step. Any permutation of this decreasing Δ_j elimination order will reduce the exploration time. The worst-case total cost is hence upper-bounded by:

$$\begin{aligned}
&\frac{K(K-1)}{2} \cdot s_{K-1} && + \\
&\frac{(K-1)(K-2)}{2} \cdot (s_{K-2} - s_{K-1}) && + \\
&\dots \\
&\frac{(j+1)j}{2} \cdot (s_j - s_{j+1}) && + \\
&\dots \\
&(\varepsilon+1) \cdot s_{\varepsilon+1}
\end{aligned}$$

If we reorder the sum we have:

$$\begin{aligned}
&\left(\frac{K(K-1)}{2} - \frac{(K-1)(K-2)}{2}\right) \cdot s_{K-1} && + \\
&\left(\frac{(K-1)(K-2)}{2} - \frac{(K-2)(K-3)}{2}\right) \cdot s_{K-2} && + \\
&\dots \\
&\left(\frac{(j+1)j}{2} - \frac{j(j-1)}{2}\right) \cdot s_j && + \\
&\dots \\
&(\varepsilon+1) \cdot s_{\varepsilon+1}
\end{aligned}$$

$$= \sum_{j=\varepsilon+1}^{K-1} j \cdot s_j$$

hence the result. $\square$

B. Additional content

B.1. A note about Explore then Exploit

The "budgeted" extension of PAC learning we introduced in Definition 1 was motivated by the *explore-then-exploit* strategy proposed in (Yue et al., 2012, section 4). When the horizon is known in advance, this strategy allows us to reduce any good exploration algorithm mechanically into a good regret minimization algorithm.

Let T be a given time horizon, a "budgeted PAC" exploration algorithm EXPLORE() is used to guess the best decision with precision $\varepsilon := 0$ and confidence $\delta := 1/T$ in less than T samples steps. Let $\hat{t}$ be the effective termination time of this algorithm ($\hat{t}$ is a random variable), let $\hat{T}(\varepsilon, \delta, N, T) \leqslant T$ be an upper bound for $\hat{t}$ (this one is not a random variable), and let $\hat{d}_s$ be the decision chosen after s sampling steps. If $\hat{t} < T$, we enter an exploit phase by repeatedly choosing the decision $\hat{d}_{\hat{t}}$. The main process terminates after T sampling rounds:

Algorithm 4 Explore then exploit

$\hat{d}_{\hat{t}} :=$ EXPLORE($\varepsilon = 0, \delta = 1/T, T$)
for $s := (\hat{t} + 1)$ **to** T **do**
  Play $\hat{d}_{\hat{t}}$
end for

The *regret* or *opportunity loss* is the difference of utility between the best strategy and a given strategy:

$$r_s = U(d^*, \boldsymbol{\mu}) - U(\hat{d}_s, \boldsymbol{\mu}) \ . \tag{21}$$

This value is a random variable and the *expected cumulative regret* at horizon T is defined by:

$$\mathbb{E}[R_T] = \mathbb{E}\left[\sum_{s=1}^{T} r_s\right] \ .$$

We split this sum into exploration and exploitation phases:

$$\mathbb{E}[R_T] = \mathbb{E}\left[\sum_{s=1}^{\hat{t}} r_s + \sum_{s=\hat{t}+1}^{T} r_s\right] \ .$$

If the utility is upper-bounded by 1, we can upper-bound the exploration regret by $\hat{T} + 1$:

$$\begin{aligned} \mathbb{E}[R_T] &\leqslant \hat{T} + (T - \hat{T}) \cdot \mathbb{E}[r_s] \\ &\leqslant \hat{T} + (T - \hat{T}) \cdot \mathbb{P}[\hat{d} \neq d^*] \\ &\leqslant \hat{T} + \frac{T - \hat{T}}{T} \\ &\leqslant \hat{T} + 1 \end{aligned}$$

B.2. A few implementation details

In order to obtain generic code, we made a heavy use of `C++` templates combined with the Boost libraries `uBLAS` and `Random`.

For our experiments, we encoded the confidence set $\mathcal{H}$ as a product of confidence intervals:

$$\mathcal{H} = \mathcal{F} \cap \bigtimes_i [a_i, b_i] \ .$$

With this encoding, the intersection step 10 of Algorithm 1 is performed by updating a_i and b_i:

$$[a_i, b_i] := [\max\{a_i, \hat{\mu}_i - c(t_i)\}, \min\{b_i, \hat{\mu}_i + c(t_i)\}]$$

Our implementation of Algorithm 2 uses a crude rejection sampling in order to sample from feasible set $\mathcal{F}$ on steps 5 and 9. Most examples we considered in our simulations use simple feasible sets, but in order to avoid infinite loops we added a security guard: if no instance is found after more than M trials, the features in consideration are added directly to $\mathcal{S}$. A side-effect of this security guard is to fall back to round-robin sampling when the proposed set $\mathcal{F}$ is empty or too flat.

B.3. An example to illustrate the formal Copeland independence predicate

In (8), the residual predicate $\mathbf{Cop}(\mathcal{H},(i,j))$ is used to test independence for parameters like $x_{3,4}$ in the following undetermined preference matrix:

$$\begin{array}{c|c|c|c|c|c|l} 1 & & 1 & 1 & 1 & 0 & U_{Cop}(1,\mathbf{x}) = 3 \\ 2 & 0 & & 1 & 1 & 0 & U_{Cop}(2,\mathbf{x}) = 2 \\ 3 & 0 & 0 & & \mathbf{?} & \mathbf{?} & U_{Cop}(3,\mathbf{x}) \in [0,2] \\ 4 & 0 & 0 & \mathbf{?} & & 1 & U_{Cop}(4,\mathbf{x}) \in [1,2] \\ \mathbf{5} & 1 & 1 & \mathbf{?} & 0 & & U_{Cop}(5,\mathbf{x}) \in [2,3] \end{array} \tag{22}$$

This matrix is defined from $\mathcal{H} = \times_{i<j}[a_{i,j}, b_{i,j}]$ by:

$$m_{i,j} = \begin{cases} 1 & \text{when } a_{i,j} > 1/2 \\ 0 & \text{when } b_{i,j} < 1/2 \\ \mathbf{?} & \text{when } \frac{1}{2} \in [a_{i,j}, b_{i,j}] \end{cases} \tag{23}$$

The Copeland winner decision is independent of $x_{3,4}$ because both $U_{Cop}(3,\mathbf{x})$ and $U_{Cop}(4,\mathbf{x})$ are dominated by $U_{Cop}(1,\mathbf{x})$. On the other hand, the decision is not independent of $x_{3,5}$ because $U_{Cop}(5,\mathbf{x})$ is not dominated. If we use (8) without $\mathbf{Cop}(\mathcal{H},(i,j))$ term, we only have a sufficient condition of independence.

B.4. An example to illustrate the formal Condorcet independence predicate

If the barmaid of the casino has been kind enough to tell us that our preference matrix admits a Condorcet winner, the feasible set will become the one of (10). For instance if we have the following undetermined preference matrix:

$$\begin{array}{c|c|c|c|c|c|l} 1 & & \mathbf{?} & \mathbf{?} & \mathbf{?} & 1 & U_{Cop}(1,\mathbf{x}) \in [1,4] \\ 2 & \mathbf{?} & & 0 & \mathbf{?} & \mathbf{?} & U_{Cop}(2,\mathbf{x}) \in [0,3] \\ 3 & \mathbf{?} & 1 & & 0 & \mathbf{?} & U_{Cop}(3,\mathbf{x}) \in [1,3] \\ 4 & \mathbf{?} & \mathbf{?} & 1 & & 0 & U_{Cop}(4,\mathbf{x}) \in [1,3] \\ 5 & 0 & \mathbf{?} & \mathbf{?} & 1 & & U_{Cop}(5,\mathbf{x}) \in [1,3] \end{array} \tag{24}$$

By definition of a Condorcet winner, his Copeland index is $U_{Cop}(i^*,\mathbf{x}) = K - 1 = 4$: only row 1 can satisfy this condition. Sampling will reject non-Condorcet matrices automatically but the formal independence test (8) has to be replaced by (11) in order to give a necessary condition of independence.

Predicate (11) is encoded by:

$$\vee \begin{array}{l} \left(\exists k > i,\ b_{i,k} < \frac{1}{2} \vee \exists l < i,\ a_{l,i} > \frac{1}{2}\right) \\ \left(\exists k > j,\ b_{j,k} < \frac{1}{2} \vee \exists l < j,\ a_{l,j} > \frac{1}{2}\right) \end{array} .$$

B.5. Other parametric decision problems

As mentioned in the conclusion, the **SAVAGE** algorithm allows us to deal with several variants of bandits problems as soon as we have formalized their decision functions and independence predicates. We give here the experiment we performed with two additional examples of bandits problems:

B.5.1. Argmax bandits

As mentioned in section 3.1, with the arg max decision function, the formal independence predicate is (4). With the same distribution as in Figure 2 – but without rigged arms – we obtain the performance curves of Figure 8.

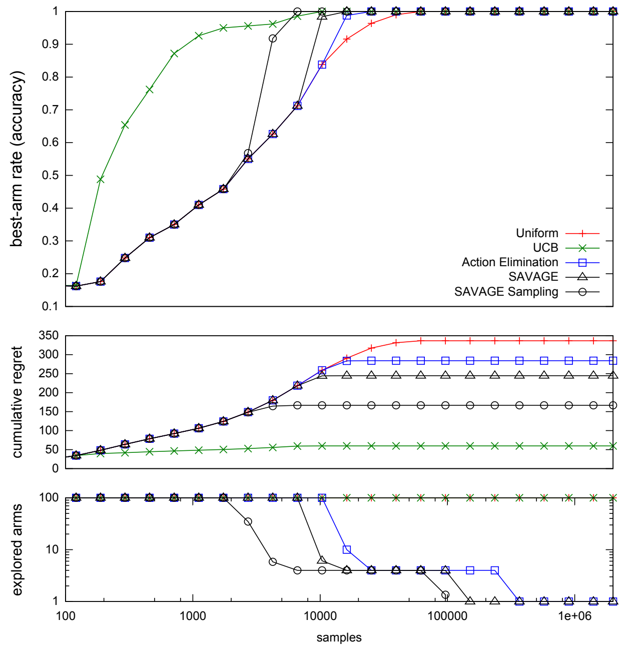

Figure 8. Behavior of the different algorithms for 1000 simulations with Figure 2 distribution and the usual arg max decision function. As expected, UCB takes the lead while the uniform policy explores too much the tail of the distribution.

It is worth noting that SAVAGE and (Even-Dar et al., 2002) action elimination behave differently even on this classical bandit setting.

B.5.2. Counting bandits

There is no need for the decision space to be the parameter space, the decision function does not even require to be defined as a utility maximization. For instance the parametric decision may be the number of parameters satisfying a given predicate in which case $\mathcal{D} = \{0, \ldots, N\}$. This is what we call *counting bandits*. On Figure 9 we give an experiment we made with $f(\boldsymbol{x}) := |\{i \mid x_i \in (0.4, 0.6)\}|$.

The real count is $d^* = f(\boldsymbol{\mu})$, and for the regret we used the error $|d^* - f(\hat{\boldsymbol{\mu}})|$. The independence predicate is:

$$\textbf{IndepTest}_f(\mathcal{H}(\mathbf{t}), i) = (b_i < 0.4 \vee 0.6 < a_i) \vee (0.4 < a_i \wedge b_i < 0.6) \ . \tag{25}$$

B.6. Additional dueling bandits experiments

The simulation of Figure 10 is the same as the one of Figure 6 but with $K = 50$. Figure 11 is a simulation we made with the same preference matrix as in (Yue & Joachims, 2011). For the regret, we used an utility defined by $1 - \mu_{i*,i}$. A main difference with (Yue & Joachims, 2011) remains: we do not assume here the decision to be the explored arm. As explained at the end of Section 3, it is difficult to outperform a uniform exploration strategy on the long run with these kind of "flat" preference matrices.

B.7. Decision functions visualization attempt

We give here the results of some side-experiments we performed in order to map decision functions. These visualizations are of course extremely distorted but they allows us illustrate the principle of generic exploration: we navigate in fog, but we have the map.

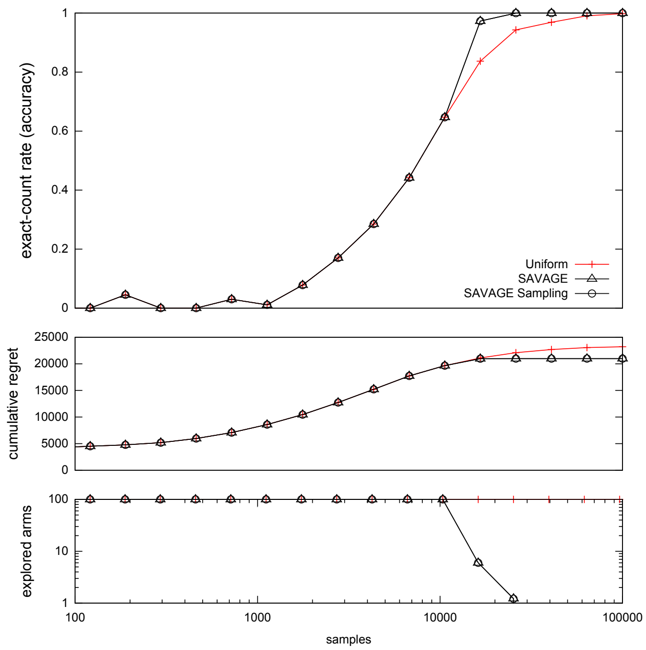

Figure 9. The distribution of Figure 2 with a counting decision function.

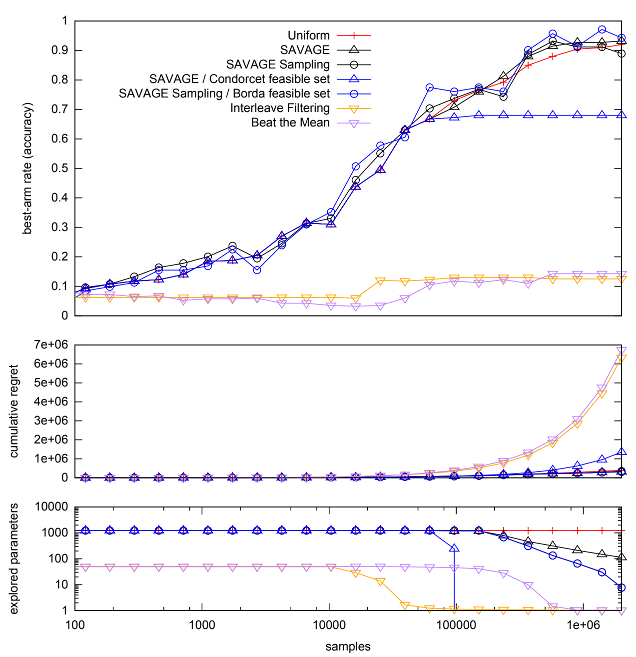

Figure 10. Same random setting as in Figure 6 with $K = 50$ instead of 30.

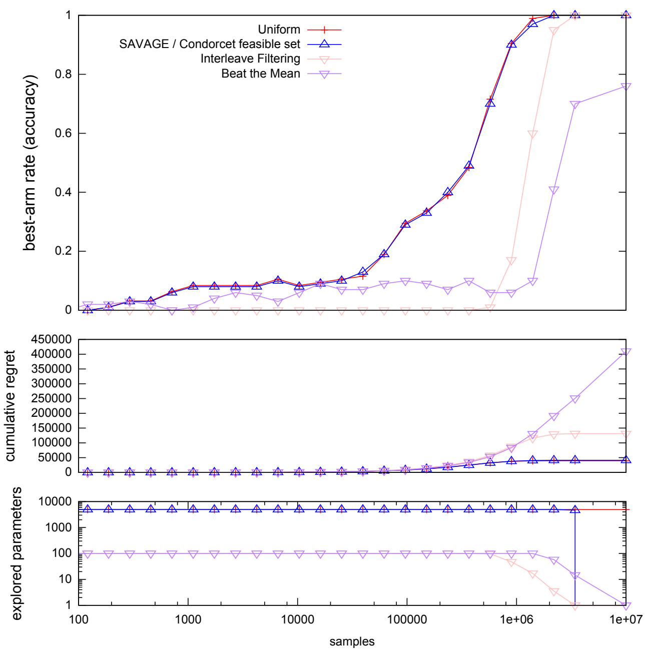

Figure 11. An experiment with $K = 100$ and a flat Condorcet preference matrix ($\mu_{i,j} = 0.6$ for all $i < j$).

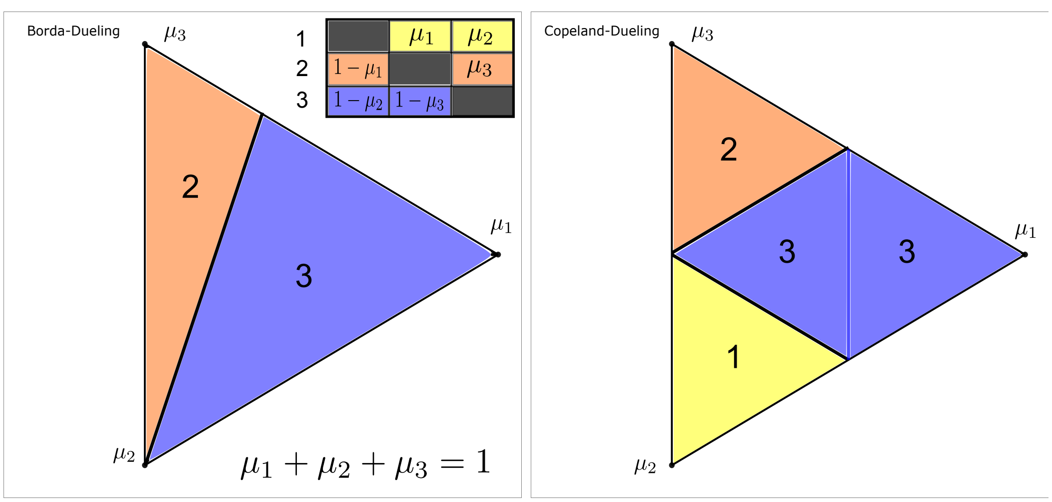

Figure 12. A 2D slice of the 3-dueling bandits decision space with resp. Borda and Copeland winning conditions. The 3×3-preference matrix is entirely defined by 3 parameters $p_{1,2} = \mu_1$, $p_{1,3} = \mu_2$, and $p_{2,3} = \mu_3$ (see Section 4). The Borda winner is the row i which maximizes the sum $\sum_j p_{i,j}$ while the Copeland winner is the one which maximizes the number of one-to-one "victories" $\sum_j [p_{i,j} > 0.5]$. In case of tie, the lowest index is chosen.

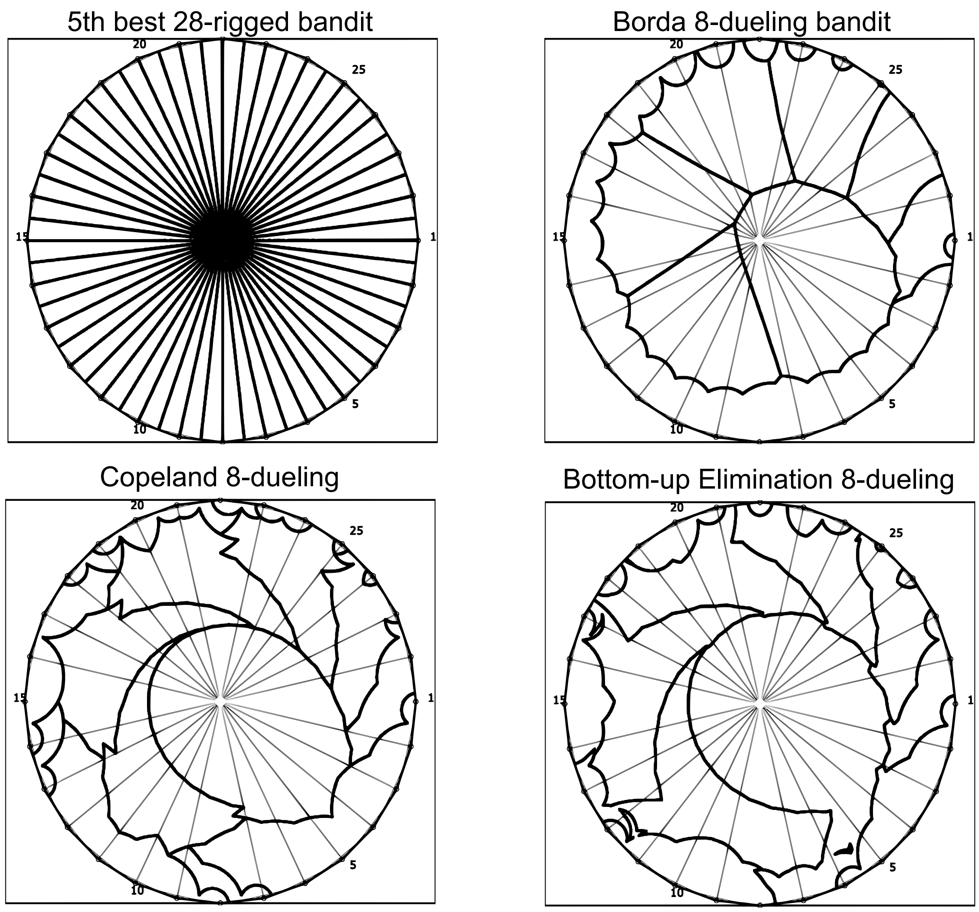

Figure 13. Four examples of decision boundaries plotted on the same surface embedded into a decision space of dimension 28: each time we cross a bold line, the "best arm" decision changes. The top left figure show a rigged bandits with 28 arms. The three other examples shape the decision boundaries obtained with different election criteria (with a preference matrix of size 8 we have 28 free parameters).