Contextual Dueling Bandits

Miroslav Dudík   MDUDIK@MICROSOFT.COM
Microsoft Research, New York, NY USA

Katja Hofmann   KATJA.HOFMANN@MICROSOFT.COM
Microsoft Research, Cambridge, United Kingdom

Robert E. Schapire[*]   SCHAPIRE@MICROSOFT.COM
Microsoft Research, New York, NY USA

Aleksandrs Slivkins   SLIVKINS@MICROSOFT.COM
Microsoft Research, New York, NY USA

Masrour Zoghi[†]   M.ZOGHI@UVA.NL
University of Amsterdam, Amsterdam, The Netherlands

Abstract

We consider the problem of learning to choose actions using contextual information when provided with limited feedback in the form of relative pairwise comparisons. We study this problem in the dueling-bandits framework of Yue et al. (2009), which we extend to incorporate context. Roughly, the learner's goal is to find the best policy, or way of behaving, in some space of policies, although "best" is not always so clearly defined. Here, we propose a new and natural solution concept, rooted in game theory, called a *von Neumann winner*, a randomized policy that beats or ties every other policy. We show that this notion overcomes important limitations of existing solutions, particularly the Condorcet winner which has typically been used in the past, but which requires strong and often unrealistic assumptions. We then present three efficient algorithms for online learning in our setting, and for approximating a von Neumann winner from batch-like data. The first of these algorithms achieves particularly low regret, even when data is adversarial, although its time and space requirements are linear in the size of the policy space. The other two algorithms require time and space only logarithmic in the size of the policy space when provided access to an oracle for solving classification problems on the space.

Keywords: contextual dueling bandits, online learning, bandit algorithms, game theory.

1. Introduction

We study how to learn to act based on contextual information when provided only with partial, relative feedback. This problem naturally arises in information retrieval (IR) and recommender systems, where the user feedback is considerably more reliable when interpreted as relative comparisons rather than absolute labels (Radlinski et al., 2008). For instance, in web search, for a particular query, the IR system may have several candidate rankings of documents that could be presented, with the best option being dependent upon the specific user. By presenting a mix or interleaving of two of the candidate rankings and observing the user's response (Chapelle et al., 2012; Hofmann et al., 2013), it is possible for such a system to get feedback about user preferences.

---

[*] On leave from Princeton University.

[†] Part of this research was conducted during an internship with Microsoft Research.

However, this feedback is partial since it is only with respect to the two rankings that were chosen, and it is relative since it only tells which of the two rankings is preferred to the other.

The *dueling-bandits problem* of Yue et al. (2009) formalizes this setting. Abstractly, the learner is repeatedly faced with a set of possible actions, and may select two of these actions to face off in a *duel* whose stochastically determined winner is then revealed. Through such experimentation, the learner attempts to find the "best" of the actions.

In this paper, we focus on the *contextual dueling bandit* setting, where context can provide information that helps identify the best action. For instance, in the example above, the actions may be the candidate rankings to choose among, and the context may be additional information about the user or query that might help in choosing the best ranking. The learner's goal now is to find a good *policy*, a rule for choosing actions based on context.

Similar to prior work on contextual (non-dueling) bandits (Auer et al., 2002; Langford and Zhang, 2007; Dudík et al., 2011; Agarwal et al., 2014), we propose a setting in which the learner has access to a space of policies Π, with the goal of performing as well as the "best" in the space. This space plays a role analogous to the hypothesis space in supervised learning. It will typically be extremely large or even infinite. We therefore explicitly aim for methods that will be applicable when this is the case.

Merely defining the precise goal of learning can be problematic in such a relative-feedback setting. When rewards are absolute, the best policy in Π is clearly and easily defined as the one that achieves the highest expected reward, because, by such an absolute measure, this policy beats every other policy. In a relative-feedback setting, since we have a means of obtaining pairwise comparisons between actions or policies, we might aim to choose the policy in Π that (on average) beats every other policy in the class in such head-to-head competitions. Most previous work on dueling bandits (Yue et al., 2009; Yue and Joachims, 2011; Urvoy et al., 2013; Zoghi et al., 2014b) has in fact explicitly or implicitly assumed that such a *Condorcet winner* exists. But there are good reasons to doubt such a strong assumption, particularly when working with large and rich policy spaces. There are numerous examples, even in natural situations, where this assumption (and more generally, transitivity among policies) is known to fail (see, for instance, Gardner, 1970; Zoghi et al., 2014a). Indeed, the preferences of a population of users do not need to be transitive, even if each individual user has transitive preferences.

In this paper, we seek to improve the dueling bandits techniques in two respects. First, we seek to relax the modeling restrictions on which previous methods have depended so as to develop methods that are more generally applicable. Second, we seek to achieve a similar level of flexibility in the design of policies as for supervised learning algorithms.

Contributions. Our first contribution (in Section 2) is the introduction of a new solution concept, called the *von Neumann winner*, which is based on a game-theoretic interpretation. Like a Condorcet winner, when facing any other policy in a duel, a von Neumann winner has at least a 50% chance of winning; in this sense, a von Neumann winner is at least as good as every policy in the space. On the other hand, a von Neumann winner is *always* guaranteed to exist, without any extraneous assumptions. This guarantee is made possible by allowing policies to be selected in a *randomized* fashion, as is quite natural in such a learning setting.

With the goal of learning clarified, we turn to algorithms. As a warm-up, in Section 5, we give a fully online algorithm in which two copies of the Exp4.P multi-armed bandit algorithm (Beygelzimer et al., 2011) are run against one another (using a "sparring" approach previously suggested

by Ailon et al., 2014). Although yielding good regret, this algorithm requires time and space linear in $|\Pi|$, which is impractical in most realistic settings where we would expect Π to be enormous.

To address this difficulty, we propose an approach used previously in other works on contextual bandits (Langford and Zhang, 2007; Dudík et al., 2011; Agarwal et al., 2014). Specifically, we assume that we have access to a *classification oracle* for our policy class that can find the minimum-cost policy in Π when given the cost of each action on each of a sequence of contexts. In fact, an ordinary cost-sensitive, multiclass classification learning algorithm can be used for this purpose, which suggests that, practically, this may be a reasonable and natural assumption.

We then consider techniques for constructing a von Neumann winner from empirical exploration data. (Although we focus on a batch-like setting, the resulting algorithms can be used online as well.) We analyze the statistical efficiency of this approach in Section 6. In Sections 7 and 8, we give two polynomial-time algorithms for computing an approximate von Neumann winner from data: one based on Kalai and Vempala's Follow-the-Perturbed-Leader algorithm (2003), and the other based on projected gradient ascent as studied by Zinkevich (2003). These techniques yield learning algorithms that approximate or perform as well as the von Neumann winner, using data, time, and space that only depend logarithmically on the cardinality of the space Π, and therefore, are applicable even with huge policy spaces.

Other related work. Numerous algorithms have been proposed for the (non-contextual) dueling bandits problem: Interleaved Filter (Yue et al., 2009); Beat the Mean (BTM) (Yue and Joachims, 2011); Sensitivity Analysis of VAriables for Generic Exploration (SAVAGE) (Urvoy et al., 2013); Relative Confidence Sampling (Zoghi et al., 2014a); Relative Upper Confidence Bound (RUCB) (Zoghi et al., 2014b); Doubler, MultiSBM and Sparring (Ailon et al., 2014) and mergeRUCB (Zoghi et al., 2015b). These methods impose various constraints on the problem at hand, ranging from the requirement that it arise from an underlying utility function (e.g., MultiSBM) to no constraint at all (e.g., SAVAGE); they mainly provide regret bounds that are logarithmic in the number of rounds, and at least linear in the number of actions. In principle, these methods could be applied to contextual dueling bandits by treating policies as actions. But this would lead to regret at least linear in the number of policies which is far worse than the logarithmic bounds obtained in this paper.

The method that is the most closely related to our work is Dueling Bandit Gradient Descent (DBGD) (Yue and Joachims, 2009), a policy gradient method that iteratively improves upon the current policy by conducting comparisons with nearby policies, assuming that the policy space comes equipped with a distance metric, and incrementally adapting the policy if a better alternative is encountered. As with all local optimization methods, DBGD imposes a convexity assumption on the dueling bandit problem for its performance guarantee: the dueling bandit problem is assumed to arise from the noisy observations of an underlying convex objective function. In this paper, we both relax the assumptions imposed by DBGD and improve upon the regret bound.

2. Dueling bandits and the von Neumann winner

In the *dueling bandits problem* (Yue et al., 2009), the learner has access to K possible *actions*, $1, \ldots, K$, and attempts to determine the "best" action through repeated stochastic pairwise comparisons of actions, called *duels*. Thus, at each time step, the learner chooses a pair of actions (a, b) for a duel; the outcome of the duel is $+1$ if a wins, and -1 if b wins. The (unknown) expected value of this outcome is denoted $P(a, b)$, and is assumed to depend only on the selected pair (a, b). In other words, the probability that a beats b in a duel is $(P(a, b) + 1)/2$, and the two actions are exactly

evenly matched if $P(a, b) = 0$. We say that a *beats* b to mean that the chance of a winning a duel with b is strictly greater than $1/2$; similarly, a *ties* b if this probability is exactly $1/2$.

The $K \times K$ matrix $\mathbf{P}$ of all such expectations $P(a, b)$ is called the *preference matrix*.[1] This matrix is initially unknown to the learner, but can be discovered bit-by-bit through experimentation. We assume, of course, that all of the entries of $\mathbf{P}$ are in $[-1, +1]$, and furthermore, that $\mathbf{P}$ is *skew-symmetric*, meaning that $\mathbf{P}^\top = -\mathbf{P}$ so that a duel (b, a) is equivalent to (the negation of) a duel (a, b). (This also implies $P(a, a) = 0$ for every action a, as is natural.) Other than this, we strenuously avoid making any assumptions in the current work about the matrix $\mathbf{P}$. For instance, we do *not* make any assumptions regarding transitivity among the various actions.

In such a relative-feedback setting, the "best" action is not always well-defined because there is no measure of the absolute quality of actions. Existing work typically assumes the existence of a *Condorcet winner* (Urvoy et al., 2013; Zoghi et al., 2014b), that is, an action a^* that beats every other action $a \neq a^*$. This is a very natural definition from a preference learning perspective, since a^* is indeed preferred to every other action. However, it has been shown that dueling bandit problems without Condorcet winners arise regularly in practice (Zoghi et al., 2014a).[2]

Although there is no guarantee of a *single* action beating all others, the situation changes considerably if we simply allow actions to be selected in a *randomized* fashion. With this natural relaxation, the problem of non-existence entirely vanishes. Thus, the idea is to find a probability vector $\mathbf{w}$ in Δ_K (where Δ_K is the simplex of vectors in $[0, 1]^K$ whose entries sum to 1) such that

$$\sum_{a=1}^{K} w(a)P(a, b) \geq 0 \quad \text{for all actions } b. \tag{1}$$

In words, for every action b, if a is selected randomly according to distribution $\mathbf{w}$, then the chance of beating b in a duel is at least $1/2$. (Note that this property implies that the same will be true if b is itself selected in a randomized way.) A distribution $\mathbf{w}$ with this property is said to be a *von Neumann winner* for the preference matrix $\mathbf{P}$.

As the name reflects, this notion is intimately connected to a game-theoretic interpretation. Indeed, we can view preference matrix $\mathbf{P}$ as describing a *zero-sum matrix game*. In such a game, the two players simultaneously choose distributions (or *mixed strategies*) $\mathbf{w}$ and $\mathbf{u}$ over rows and columns, respectively, yielding a gain to the row player of $\mathbf{w}^\top \mathbf{P} \mathbf{u}$. According to von Neumann's celebrated minmax theorem, for any matrix $\mathbf{P}$,

$$\max_{\mathbf{w} \in \Delta_K} \min_{\mathbf{u} \in \Delta_K} \mathbf{w}^\top \mathbf{P} \mathbf{u} = \min_{\mathbf{u} \in \Delta_K} \max_{\mathbf{w} \in \Delta_K} \mathbf{w}^\top \mathbf{P} \mathbf{u},$$

the common value being the *value* of the game $\mathbf{P}$. A *maxmin strategy* $\mathbf{w}$ or a *minmax strategy* $\mathbf{u}$ is one realizing the $\max$ or $\min$ on the left- or right-hand side of this equality, respectively. Finding these strategies is called *solving the game*.

We have assumed that the matrix $\mathbf{P}$ is itself skew-symmetric, so the game it describes is a *symmetric game*. Such games are known to have value exactly equal to zero (see, for instance, Owen, 1995, Theorem II.6.2). Working through definitions, this means that $\mathbf{w}$ is a maxmin strategy if and only if $\min_{\mathbf{u} \in \Delta_K} \mathbf{w}^\top \mathbf{P} \mathbf{u} \geq 0$. But this is exactly equivalent to Eq. (1). Therefore:

Proposition 1 *A probability vector* $\mathbf{w} \in \Delta_K$ *is a von Neumann winner for preference matrix* $\mathbf{P}$ *if and only if it is a maxmin strategy for the game* $\mathbf{P}$. *Consequently, every preference matrix* $\mathbf{P}$ *has a von Neumann winner.*

---

1. In the literature, the preference matrix $\mathbf{P}$ often refers to the matrix of probabilities $(P(a, b) + 1)/2$. With our modification, $P(a, b)$ becomes anti-symmetric around 0, which simplifies the arguments considerably.

2. See also Appendix A for more compelling evidence that this is indeed the case.

Before continuing, we briefly mention some of the other solution concepts that have been proposed to remedy the potential non-existence of a Condorcet winner (Schulze, 2011). Two of these are the Borda winner, the action that has the highest probability of winning a duel against a uniformly random action; and the Copeland winner, the action that wins the most pairwise comparisons (Urvoy et al., 2013). Both of these fail the *independence of clones criterion* (Schulze, 2011), meaning that adding multiple identical copies of an action can change the Borda or Copeland winner. This criterion is particularly crucial in a dueling bandit setting because a given policy class may contain many identical policies. In contrast, the von Neumann winner performs at least as well as any individual policy, and is thus unaffected by the presence or absence of clones. See Appendix B for a more detailed discussion. Note that if there does happen to exist a Condorcet winner, then it will also be the unique von Neumann winner.

3. Incorporating context

Next, we consider how the preceding development can be extended to a much more realistic setting in which the best way of acting may depend on additional, observable information, called *context*. Thus, prior to choosing actions, the learner is allowed to observe some value x, the context, selected by Nature from some unspecified space $\mathcal{X}$. For instance, x might be a feature-vector description of a web user. In this setting, the preference matrix is no longer static; rather, which actions are better than which others now varies and depends on the context, which therefore must be taken into account to fully optimize the choice of actions.

Formally, we assume that on every round t of the learning process, a context x_t and preference matrix $\mathbf{P}_t$ are chosen by Nature. The context x_t is revealed to the learner, but the preference matrix $\mathbf{P}_t$ remains hidden. Based on x_t, the learner selects two actions (a_t, b_t) for a duel, whose outcome has expectation determined by the current (hidden) preference matrix $\mathbf{P}_t$ in the usual way. Except where noted otherwise, in this paper, we always assume that each pair $(x_t, \mathbf{P}_t)$ is chosen at random according to independent draws from some unknown joint distribution $\mathcal{D}$.

The goal is to determine which action to select as a function of the context. Such a mapping π from contexts x to actions a is called a *policy*. Typically, we are working with policies of a particular form, that is, from some *policy space* Π. For instance, this space might represent the set of all decision trees. For simplicity, we assume that Π has finite cardinality. However, we generally think of Π as an extremely large space, exponential in any reasonable measure of complexity.

The notion of von Neumann winner (as well as other concepts, like Condorcet winner) can be extended to incorporate context essentially by reducing to the non-contextual setting. We can regard each policy π as a "meta-action," and define a $|\Pi| \times |\Pi|$ preference matrix $\mathbf{M}$ over these meta-actions. Thus, the rows and columns of $\mathbf{M}$ are each indexed by policies in Π, and

$$M(\pi, \rho) = \mathbb{E}_{(x, \mathbf{P}) \sim \mathcal{D}} \left[P(\pi(x), \rho(x)) \right]. \tag{2}$$

This quantity is the expected outcome when a "meta-duel" is held between the two *policies* π and ρ, whose stochastic outcome is determined by randomly selecting $(x, \mathbf{P}) \sim \mathcal{D}$, and then holding an ordinary duel on $\mathbf{P}$ between the actions $\pi(x)$ and $\rho(x)$. This huge matrix $\mathbf{M}$ thus encodes the probability of any policy beating any other policy in a duel.

We can now define von Neumann winner in the contextual setting to be an (ordinary) von Neumann winner for the matrix $\mathbf{M}$ (regarded here as a kind of "meta-preference-matrix"). Thus, unraveling definitions, a (contextual) von Neumann winner is a probability distribution $\mathbf{W}$ over *policies*

such that for every opposing policy ρ, if π is chosen at random from $\mathbf{W}$, then the probability that π beats ρ in a duel is at least $1/2$. That is, the *randomized* policy defined by $\mathbf{W}$ beats or ties every policy in the space Π. By Proposition 1 (applied to $\mathbf{M}$), such a von Neumann winner must exist.

For the rest of the paper, we study how to compute (or approximate) contextual von Neumann winners. Of course, because the space Π and corresponding matrix $\mathbf{M}$ are both gigantic, this will present significant computational challenges.

4. Learning scenarios

We consider two possible learning scenarios.

In the simpler of these, called *explore-then-exploit*, we suppose that the learner is allowed to explore for some number of rounds m (where, as described above, on each round, the learner is presented with a random context and permitted to run and observe the outcome of a duel between a pair of actions of its choosing). At the end of these m rounds, the learner outputs a distribution $\hat{\mathbf{W}}$ over policies in Π. The learner's goal is to produce $\hat{\mathbf{W}}$ which is an *ε-approximate von Neumann winner*, that is, for which

$$\min_{\mathbf{U} \in \Delta_{|\Pi|}} \hat{\mathbf{W}}^{\top} \mathbf{M} \mathbf{U} \geq -\varepsilon$$

for some small $\varepsilon > 0$. In other words, for all $\pi \in \Pi$, $\hat{\mathbf{W}}$ should beat π with probability at least $1/2 - \varepsilon/2$. Naturally, m should be "reasonable" as a function ε. This setting is almost like learning from a passively selected batch of training examples, except that the learner has an active role in selecting which actions to play in each duel.

In the alternative *full-explore-exploit* setting, learning occurs in a fully online manner across T rounds (in the manner described earlier), with performance measured using some notion of *regret*. In this paper, where we are working with policies and changing preference matrices, we propose to define regret to be

$$\max_{\pi \in \Pi} \frac{1}{2} \sum_{t=1}^{T} \left[P_t(\pi(x_t), a_t) + P_t(\pi(x_t), b_t) \right]. \tag{3}$$

If we can find an algorithm for which this regret is $o(T)$, then eventually the algorithm selects actions (a_t, b_t) which cannot be beaten by any other policy $\pi \in \Pi$.

In the standard dueling-bandits setting with a static preference matrix, a seemingly different definition of regret was used by Yue et al. (2009) in terms of an assumed Condorcet winner. However, when specialized to their setting, and when provided with their same assumptions, their definition can be shown to be equivalent (up to constant factors) to Eq. (3).

5. Sparring Exp4.P

Our goal then is to find, approximate or perform as well as a von Neumann winner, which, as we have seen, is a maxmin strategy for a particular game. Under this interpretation, it becomes especially natural to use ordinary no-regret learning algorithms as players of this game since it is known that such algorithms, when properly configured for this purpose, will converge to maxmin or minmax strategies (Freund and Schapire, 1999). The idea is simply to run two independent copies of such an algorithm against one another. Such a "sparring" approach was previously proposed for dueling bandits by Ailon et al. (2014), though without details, and not in the contextual setting.

We consider using the multi-armed bandit algorithm Exp4.P (Beygelzimer et al., 2011) for this purpose in the full-explore-exploit setting. Exp4.P is well-suited since it is designed to work with partial information as in our bandit setting, and since it can handle the kind of adversarially generated data that arises unavoidably when playing a game. It also is designed to work with policies in a contextual setting like ours (or, more generally, to accept the advice of "experts").

The learning setting for Exp4.P is as follows (somewhat, but straightforwardly, modified for our present purposes). There are K possible actions, $1, \ldots, K$, and a finite space Π of policies. On each round $t = 1, \ldots, T$, an adversary chooses and reveals a context x_t, and also chooses, but does *not* reveal rewards $r_t(1), \ldots, r_t(K) \in [-1, +1]$ for each of the K actions. The learner then selects an action a_t, and receives the revealed reward $r_t(a_t)$. The learner's total reward is thus $G_A = \sum_{t=1}^{T} r_t(a_t)$, while the reward of each policy π is $G_\pi = \sum_{t=1}^{T} r_t(\pi(x_t))$. The learner's goal is to receive reward close to that of the best policy. Beygelzimer et al. (2011) prove that (subject to very benign conditions) with probability at least $1 - \delta$, Exp4.P achieves reward

$$G_A \geq \max_{\pi \in \Pi} G_\pi - 12\sqrt{KT \ln(|\Pi|/\delta)}. \tag{4}$$

(This holds for any $\delta > 0$; the δ is passed as a parameter to the algorithm.)

For contextual dueling bandits, we run two separate copies of Exp4.P which are played against one another; let us call them row-Exp and column-Exp. We use the same actions, contexts, and policies for the two copies as for the original problem. On each round t, Nature chooses a context x_t and a preference matrix $\mathbf{P}_t$. The context (but not the preference matrix) is revealed to row-Exp and column-Exp, which select actions a_t and b_t, respectively. A duel is then held between these two actions; the outcome r is passed as feedback to row-Exp (for its chosen action a_t), and its negation $-r$ is similarly passed to column-Exp. We call this algorithm SparringEXP4.P.

Theorem 2 *Consider K actions, policy space Π, and time horizon T. Fix parameter $\delta > 0$. Then with probability at least $1 - \delta$, SparringEXP4.P achieves regret at most $O(\sqrt{KT \ln(|\Pi|/\delta)})$.*

The proof is in Appendix C. This result holds also for an adversarial environment in which the pairs $(x_t, \mathbf{P}_t)$ are selected by an adversary rather than at random. Also, we can adapt this algorithm for explore-then-exploit learning using the following standard technique for online-to-batch conversion. Run SparringEXP4.P for m exploration rounds. In each round i, row-Exp internally computes a distribution $\mathbf{w}_i$ over policies. Then $\overline{\mathbf{w}} = (1/m) \sum_{i=1}^{m} \mathbf{w}_i$ is an ε-approximate von Neumann winner where $\varepsilon = O(\sqrt{K \ln(|\Pi|/\delta)/m})$.

Although yielding very good regret bounds and handling adversaries, this approach requires time and space proportional to $|\Pi|$, and is therefore not practical for extremely large policy spaces.

6. Explore-then-exploit algorithms with a classification oracle

We next begin a development that will lead to efficient methods (in terms of time, space and data) for handling even extremely large policy spaces, under a particular assumption discussed below. We describe a general approach for exploration, for using the collected data to find a statistically sound solution, and for reducing the problem that must be solved to a more tractable form.

We focus mainly on the explore-then-exploit problem. Thus, we have m exploration rounds, and on each round i, a pair $(x_i, \mathbf{P}_i)$ is selected at random, and the learner is permitted to choose and observe the outcome of a single duel (a_i, b_i). Although x_i is observed, $\mathbf{P}_i$ is not. Here, we propose

a simple exploration strategy, called *uniform exploration*, in which each dueling pair (a_i, b_i) is selected uniformly at random. Let r_i be the resulting observed outcome. Based on these, the learner can obtain a noisy but unbiased version of the hidden preference matrix $\mathbf{P}_i$. Specifically, let us define a matrix $\hat{\mathbf{P}}_i$ where $\hat{P}_i(a_i, b_i) = K^2 r_i$, and all other entries $\hat{P}_i(a, b)$ are set to zero. It can be verified that the expected value of each entry $\hat{P}_i(a, b)$ is exactly $P_i(a, b)$; that is, $\mathbb{E}[\hat{\mathbf{P}}_i \mid \mathbf{P}_i] = \mathbf{P}_i$.

In Appendix D, we extend our setting to an arbitrary unbiased estimator $\hat{\mathbf{P}}_i$ of $\mathbf{P}_i$, and in particular to an arbitrary exploration strategy that does not change adaptively over time. This extension is parameterized by upper bounds on the absolute value and the variance of $\hat{P}_i(a, b)$ for all rounds i and all actions (a, b). For uniform exploration, both upper bounds are K^2.

While non-adaptive exploration strategies usually lead to suboptimal statistical performance, they are often preferable in practice. This is because in large-scale industrial applications the existing infrastructure is often insufficient to support a feedback loop that would update the exploration strategy adaptively over time, and upgrading the infrastructure may be infeasible in the near term.

Statistical guarantees. With these noisy versions of the empirical preference matrices, we can estimate the expected outcome in a "meta-duel" between two policies π and ρ, that is, an entry of the matrix $\mathbf{M}$ defined in Eq. (2). In particular, let

$$\hat{M}(\pi, \rho) = \frac{1}{m} \sum_{i=1}^{m} \hat{P}_i(\pi(x_i), \rho(x_i)). \tag{5}$$

Then the expected value of this quantity is $M(\pi, \rho)$, the corresponding entry of $\mathbf{M}$. Moreover, using Bernstein's inequality and the union bound, we can show that, with probability at least $1 - \delta$,

$$\left| \hat{M}(\pi, \rho) - M(\pi, \rho) \right| \leq \varepsilon' \quad \text{for all } (\pi, \rho) \in \Pi \times \Pi, \tag{6}$$

where $\varepsilon' = O\big(K\sqrt{\ln(|\Pi|/\delta)/m}\big)$. [3] Thus, although huge, the matrix $\mathbf{M}$ is well-approximated by the matrix $\hat{\mathbf{M}}$ using only a moderately sized sample. In fact, to find an approximate maxmin strategy for $\mathbf{M}$ it suffices to find one for $\hat{\mathbf{M}}$, which will be the approach taken by our algorithms.

Lemma 3 *Given the set-up above, suppose that Eq. (6) holds (as will be the case with probability at least $1 - \delta$), and suppose further that $\hat{\mathbf{W}} \in \Delta_{|\Pi|}$ is a probability vector for which*

$$\min_{\mathbf{U} \in \Delta_{|\Pi|}} \hat{\mathbf{W}}^\top \hat{\mathbf{M}} \mathbf{U} \geq \max_{\mathbf{W} \in \Delta_{|\Pi|}} \min_{\mathbf{U} \in \Delta_{|\Pi|}} \mathbf{W}^\top \hat{\mathbf{M}} \mathbf{U} - \varepsilon.$$

Then $\hat{\mathbf{W}}$ is a $(2\varepsilon' + \varepsilon)$-approximate von Neumann winner for $\mathbf{M}$.

A more compact version of the problem. Our aim now is to find an approximate maxmin strategy for the matrix $\hat{\mathbf{M}}$. Although this matrix is gigantic in both dimensions, by leveraging how it was constructed from only a small number of empirical observations, we can re-express the problem in a far more compact form. To this end, let us define, for each policy $\pi \in \Pi$, a *policy vector* $\mathbf{v}_\pi \in \mathbb{R}^{mK}$ that encodes the behavior of π on the exploration data. For readability, although a vector, we index entries of $\mathbf{v}_\pi$ by pairs (i, a), where i is a round and a is an action, and we define

$$v_\pi(i, a) = \mathbf{1}\{\pi(x_i) = a\}/\sqrt{m}.$$

---

3. The proof for Eq. (6) and the subsequent Lemma 3 are in Appendix D.1.

Thus, $\mathbf{v}_\pi$ is broken into m length-K blocks, with block i encoding in a natural way the action selected by π on x_i. (The constant $1/\sqrt{m}$ is for normalization.)

We also define an $mK \times mK$ block-diagonal matrix $\mathbf{B}$, where the m blocks along the diagonal are exactly the $K \times K$ matrices $\hat{\mathbf{P}}_i$ described above. Formally, using the earlier indexing,

$$B((i,a),(j,b)) = \mathbf{1}\{i = j\}\hat{P}_i(a,b).$$

Working through these definitions, it can be verified that for any two policies π and ρ, the quantity $\mathbf{v}_\pi^\top \mathbf{B} \mathbf{v}_\rho$ is exactly equal to $\hat{M}(\pi, \rho)$ as defined in Eq. (5). This means that if $\mathbf{W}$ and $\mathbf{U}$ are probability vectors over Π, then

$$\mathbf{W}^\top \hat{\mathbf{M}} \mathbf{U} = \left(\sum_{\pi \in \Pi} W(\pi)\, \mathbf{v}_\pi\right)^\top \mathbf{B} \, \left(\sum_{\rho \in \Pi} U(\rho)\, \mathbf{v}_\rho\right).$$

Therefore, the problem of finding a maxmin strategy for $\hat{\mathbf{M}}$ is equivalent to solving

$$\max_{\mathbf{w} \in \mathcal{C}} \min_{\mathbf{u} \in \mathcal{C}} \mathbf{w}^\top \mathbf{B} \mathbf{u} \tag{7}$$

where $\mathcal{C}$ is the convex hull of the set of all policy vectors $\{\mathbf{v}_\pi : \pi \in \Pi\}$ (henceforth, the *policy hull*). Furthermore, a solution $\mathbf{w} \in \mathcal{C}$ is necessarily a convex combination of vectors $\mathbf{v}_\pi$, and therefore corresponds to a probability vector over policies.

The formulation given in Eq. (7) shows that $\mathbf{B}$ should itself be viewed as a game matrix, and that our remaining goal is to approximately solve this game. This matrix has the advantage of being far smaller than $\hat{\mathbf{M}}$. However, unlike a conventional matrix game, the space from which the players' vectors $\mathbf{w}$ and $\mathbf{u}$ are chosen is not the standard space of probability vectors over actions, but rather the convex hull of an exponentially large set of vectors.

Classification oracle. Our algorithms assume that the policy space Π is structured in a way that admits a certain computational operation that is quite natural in the realm of learning. Specifically, we assume the existence of a *classification oracle*. The input to this oracle is a sequence of *cost vectors* $\mathbf{c}_1, \ldots, \mathbf{c}_m$, each in $\mathbb{R}^K$, with the interpretation that $c_i(a)$ is the cost of choosing action a on context x_i. The output of the oracle is the policy in Π with minimum cost, that is,

$$\operatorname*{argmin}_{\pi \in \Pi} \sum_{i=1}^m c_i(\pi(x_i)). \tag{8}$$

Indeed, regarding the x_i's as examples, the actions a as labels or classes, and the policies π as classifiers, we see that this oracle is in fact solving an empirical, cost-sensitive, multi-class classification problem. Thus, the assumption of such an oracle is an idealization based on the numerous cases in which effective classification algorithms already exist. In practice, the policy space Π is usually defined as the space of all possible policies returned by a given classification algorithm, and we hope that our methods will be effective when using ordinary off-the-shelf classification algorithms as oracle.

Equivalently, the classification oracle can be described in terms of policy vectors. Specifically, the cost vectors above can be identified with their concatenation, a single vector $\mathbf{c} \in \mathbb{R}^{mK}$, divided naturally into m blocks. Then the problem given in Eq. (8) is the same as

$$\operatorname*{argmin}_{\mathbf{w} \in \mathcal{C}} \mathbf{c} \cdot \mathbf{w},$$

9

where the argmin is over the policy hull $\mathcal{C}$ defined above. This is because the minimum, without loss of generality, will be a policy vector $\mathbf{v}_\pi$, where π minimizes Eq. (8). Therefore, in what follows, we use expressions of this latter form to indicate an invocation of our assumed classification oracle.

Algorithms and end-to-end guarantees. We design algorithms that compute an approximate von Neumann winner $\hat{\mathbf{W}}$ by solving the optimization problem in Eq. (7). Although there exist many methods for solving such a game, the challenge here is the requirement that the solution be in the policy hull $\mathcal{C}$. As already seen in Section 5, regret minimization algorithms are a natural choice. However, most standard algorithms will not conform to this constraint. In the sections that follow, we provide two algorithms: Algorithm `SparringFPL` that builds on the Follow-the-Perturbed-Leader algorithm of Kalai and Vempala (2003), and Algorithm `ProjectedGD` that builds on online projected gradient descent methods of Zinkevich (2003).

For a given approximation quality, the performance of either algorithm is characterized by several quantities: the sufficient number of exploration rounds, the running time, the storage requirement, and the number of policies in the support of $\hat{\mathbf{W}}$. As it turns out, the key quantities are the number of exploration rounds and the number of oracle calls. We assume each oracle call returns both a policy vector and a corresponding policy, each representable using b bits.[4] The solution $\hat{\mathbf{W}}$ is specified by explicitly listing the probabilities and policies in its support.

Theorem 4 *Consider K actions and a policy class Π with b-bit representation. Fix parameters $\varepsilon, \delta > 0$. Both* `SparringFPL` *and* `ProjectedGD` *compute an ε-approximate von Neumann winner $\hat{\mathbf{W}}$ with probability $1 - \delta$ using $m = O((K^2/\varepsilon^2)\ln(|\Pi|/\delta))$ exploration rounds with uniform exploration strategy. The number of oracle calls is $N = O((K^6/\varepsilon^4)\ln(|\Pi|/\delta))$ for* `SparringFPL` *and $N = O(K^8/\varepsilon^4)$ for* `ProjectedGD`*. For both algorithms, disregarding oracle calls, the running time is $O(mKN)$, the storage requirement is $O(bN)$, and $\hat{\mathbf{W}}$ is a distribution over at most N policies.*

While `SparringFPL` is very simple and intuitive, `ProjectedGD` achieves a better number of oracle calls whenever $K \ll \sqrt{\ln(|\Pi|/\delta)}$.

Our algorithms can be used in the full-explore-exploit setting as well: after m exploration rounds with uniform exploration strategy, $\hat{\mathbf{W}}$ is computed and used in the remaining rounds for both actions. The parameter m is chosen in advance as a function of the time horizon T. The statistical performance is expressed via regret, as defined in Eq. (3). The total running time is dominated by the time to compute $\hat{\mathbf{W}}$.[5]

Theorem 5 (regret) *Consider K actions, time horizon $T \geq K$, and a policy class Π with b-bit representation. Fix a parameter $\delta > 0$. Both* `SparringFPL` *and* `ProjectedGD` *achieve regret $O(K^{2/3}\,T^{2/3}\,\Psi^{1/3})$ with probability $1 - \delta$, where $\Psi = \ln(|\Pi|/\delta)$. The number of oracle calls is $N = O(K^{10/3}\,T^{4/3}\,\Psi^{-1/3})$ for* `SparringFPL` *and $N = O(K^{16/3}\,T^{4/3}\,\Psi^{-4/3})$ for* `ProjectedGD`*. For both algorithms, disregarding oracle calls, the total running time is $O(mKN)$, the storage requirement is $O(bN)$, and the number of exploration rounds is $m = O(K^{2/3}\,T^{2/3}\,\Psi^{1/3})$.*

---

4. Often, policies are specified as a parameter vector to some algorithm that implements them. For finite classes, it is usually the case that b is roughly $O(\ln|\Pi|)$.

5. The running time to *execute* $\hat{\mathbf{W}}$ in each of the exploitation rounds (i.e., to compute the random action for a given context) is a low-order term; we omit further details from this version.

7. Solving the compact game with `SparringFPL`

Our first algorithm to solve Eq. (7), `SparringFPL`, is based on the Follow-the-Perturbed-Leader (FPL) algorithm of Kalai and Vempala (2003). FPL is designed for a standard online learning problem: Let $\mathcal{D}$ and $\mathcal{L}$ be subsets of $\mathbb{R}^{mK}$. On each round $t = 1, \ldots, N$, the learner chooses a decision vector $\mathbf{d}_t \in \mathcal{D}$, and then receives a loss vector $\boldsymbol{\ell}_t \in \mathcal{L}$. The learner's goal is to minimize its cumulative loss $\sum_{t=1}^{N} \mathbf{d}_t \cdot \boldsymbol{\ell}_t$ relative to the best possible loss using a fixed decision, that is, $\min_{\mathbf{d} \in \mathcal{D}} \sum_{t=1}^{N} \mathbf{d} \cdot \boldsymbol{\ell}_t$. FPL chooses $\mathbf{d}_t$ as the best such vector based on a slightly perturbed version of the preceding losses. Namely, letting $\mathbf{p}_t \in \mathbb{R}^{mK}$ be chosen uniformly at random from $[0, 1/\alpha]^{mK}$,

$$\mathbf{d}_t = \operatorname*{argmin}_{\mathbf{d} \in \mathcal{D}} \mathbf{d} \cdot \left(\sum_{\tau=1}^{t-1} \boldsymbol{\ell}_\tau + \mathbf{p}_t \right).$$

We solve Eq. (7) by sparring two copies of FPL, called row-FPL and column-FPL, in the fashion of a repeated game. On every round t, row-FPL uses FPL to select a vector $\mathbf{w}_t$, while column-FPL uses a different copy of FPL to select a vector $\mathbf{u}_t$. We then define the resulting loss vectors to be $-\mathbf{B}\mathbf{u}_t$ for row-FPL, and $\mathbf{B}^\top \mathbf{w}_t$ for column-FPL. Here is the complete algorithm:

- For $t = 1, \ldots, N$:
    - Choose uniform random perturbations $\mathbf{p}_t, \mathbf{q}_t$ from $[0, 1/\alpha]^{mK}$.
    - Let $\mathbf{w}_t = \operatorname{argmin}_{\mathbf{w} \in \mathcal{C}} \mathbf{w} \cdot [-\mathbf{B}(\mathbf{u}_1 + \cdots + \mathbf{u}_{t-1}) + \mathbf{p}_t]$.
    - Let $\mathbf{u}_t = \operatorname{argmin}_{\mathbf{u} \in \mathcal{C}} \mathbf{u} \cdot \left[\mathbf{B}^\top (\mathbf{w}_1 + \cdots + \mathbf{w}_{t-1}) + \mathbf{q}_t\right]$.
- Output $\overline{\mathbf{w}} = \frac{1}{N} \sum_{t=1}^{N} \mathbf{w}_t$

The argmin expressions in the algorithm are implemented using the classification oracle. The returned vector $\overline{\mathbf{w}}$ is in $\mathcal{C}$, and in fact corresponds to a uniform mixture of N policies.

In Appendix D.2, we show that to find an ε-approximate solution to Eq. (7) with probability $1 - \delta$, it suffices to use $N = O(K^4/\varepsilon^2)(m + \ln(1/\delta))$ steps of the algorithm with $\alpha = \sqrt{2/(K^4 N)}$, which in turn implies Theorems 4 and 5 for `SparringFPL`.

8. Solving the compact game with `ProjectedGD`

Our second algorithm, called `ProjectedGD`, solves Eq. (7) using online projected gradient descent methods as studied by Zinkevich (2003). The algorithm maintains a vector $\mathbf{w}_t \in \mathcal{C}$ corresponding to a strategy for the row player. On every round, a column strategy $\mathbf{u}_t \in \mathcal{C}$ is chosen that is a "best response" to $\mathbf{w}_t$. The strategy $\mathbf{w}_t$ is updated by taking a small gradient step. The resulting vector $\mathbf{z}_{t+1}$ is likely to be outside the set $\mathcal{C}$, and therefore is (approximately) projected back to $\mathcal{C}$, yielding $\mathbf{w}_{t+1}$. The algorithm is as follows:

- Choose any $\mathbf{w}_1 \in \mathcal{C}$
- For $t = 1, \ldots, N_{out}$:
    - $\mathbf{u}_t = \arg\min_{\mathbf{u} \in \mathcal{C}} \mathbf{w}_t^\top \mathbf{B}\mathbf{u}$
    - $\mathbf{z}_{t+1} = \mathbf{w}_t + \eta \mathbf{B}\mathbf{u}_t$
    - $\mathbf{w}_{t+1} = \texttt{ApproxProject}(\mathbf{z}_{t+1}, \mathbf{w}_t)$
- Output $\overline{\mathbf{w}} = \frac{1}{N_{out}} \sum_{t=1}^{N_{out}} \mathbf{w}_t$

Ideally, we would like for $\mathbf{w}_{t+1}$ to be the exact Euclidean projection of $\mathbf{z}_{t+1}$ onto $\mathcal{C}$, but instead need to settle for an approximation. For this purpose, the procedure $\texttt{ApproxProject}(\mathbf{z}, \mathbf{v}_1)$, described below, computes an approximate projection of an arbitrary vector $\mathbf{z}$ onto $\mathcal{C}$. It takes as an input a second vector $\mathbf{v}_1$ that is already in $\mathcal{C}$, and which we can think of as an initial guess at the actual projection. The quality (as an approximation) of the returned vector $\overline{\mathbf{v}}$ is allowed to depend on how close $\mathbf{v}_1$ is to $\mathbf{z}$. Specifically, we require that, for all $\mathbf{s} \in \mathcal{C}$, and a constant α specified later,

$$\|\mathbf{s} - \overline{\mathbf{v}}\|^2 \leq \|\mathbf{s} - \mathbf{z}\|^2 + \alpha \cdot \|\mathbf{v}_1 - \mathbf{z}\|. \tag{9}$$

In Appendix D.3, we show that with the parameter $\eta = 2/(L\sqrt{N_{out}})$ our algorithm finds an ε-approximate solution to Eq. (7), where $\varepsilon = 2L/\sqrt{N_{out}} + L\alpha/2$.

Computing approximate projections. It remains to describe the approximate-projection procedure $\texttt{ApproxProject}(\mathbf{z}, \mathbf{v}_1)$. Given an arbitrary vector $\mathbf{z}$ and another vector $\mathbf{v}_1 \in \mathcal{C}$, the goal of the algorithm, as in Eq. (9), can be restated as that of finding a vector $\overline{\mathbf{v}} \in \mathcal{C}$ for which

$$\min_{\mathbf{s} \in \mathcal{C}} F(\mathbf{s}, \overline{\mathbf{v}}) \geq -\alpha \cdot \|\mathbf{v}_1 - \mathbf{z}\| \tag{10}$$

where we define $F(\mathbf{s}, \mathbf{v}) = \|\mathbf{s} - \mathbf{z}\|^2 - \|\mathbf{s} - \mathbf{v}\|^2 = 2\mathbf{s} \cdot (\mathbf{v} - \mathbf{z}) + \|\mathbf{z}\|^2 - \|\mathbf{v}\|^2$. Note that F is linear in $\mathbf{s}$ (for each $\mathbf{v}$), and concave in $\mathbf{v}$ (for each $\mathbf{s}$). To ensure that Eq. (10) holds, we give an algorithm that aims to maximize the left-hand side of this inequality. (As a side note, the maximizing vector turns out to be exactly the projection of $\mathbf{z}$ onto $\mathcal{C}$, although we do not require that fact for our algorithm and analysis.)

To this end, we use an algorithm that resembles repeated play of a game in which the payoff is defined by F. The $\mathbf{s}$ player uses best response on each round, while the $\mathbf{v}$ player again uses a variant of online gradient ascent applied to the function $F(\mathbf{s}_t, \cdot)$. The algorithm takes a parameter $\nu \in (0, 1]$, and uses $\mathbf{v}_1 \in \mathcal{C}$, which was provided as an argument to $\texttt{ApproxProject}(\mathbf{z}, \mathbf{v}_1)$, as the initial vector. Here is the algorithm:

- For $t = 1, \ldots, N_{in}$:

    - $\mathbf{s}_t = \arg\min_{\mathbf{s} \in \mathcal{C}} \mathbf{s} \cdot (\mathbf{v}_t - \mathbf{z})$
    - $\mathbf{v}_{t+1} = (1 - \nu)\mathbf{v}_t + \nu \mathbf{s}_t$

- Output $\overline{\mathbf{v}} = \frac{1}{N_{in}} \sum_t \mathbf{v}_t$

Note that $\mathbf{v}_t$ is in $\mathcal{C}$ for every t (by convexity of $\mathcal{C}$), and therefore $\overline{\mathbf{v}}$ is as well.

In Appendix D.4, we show that $\texttt{ApproxProject}(\mathbf{z}, \mathbf{v}_1)$ with parameter $\nu = \|\mathbf{z} - \mathbf{v}_1\|/\sqrt{N_{in}}$ computes $\overline{\mathbf{v}}$ that satisfies Eq. (10) with $\alpha = 8/\sqrt{N_{in}}$. We optimize the choice of N_{in} and N_{out} to show that one can obtain an ε-approximate solution to Eq. (7) using only $O(K^8/\varepsilon^4)$ oracle calls. This in turn implies Theorems 4 and 5 for $\texttt{ProjectedGD}$.

Acknowledgments

References

Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert E. Schapire. Taming the monster: A fast and simple algorithm for contextual bandits. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2014.

Nir Ailon, Zohar Karnin, and Thorsten Joachims. Reducing dueling bandits to cardinal bandits. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 856–864, 2014.

Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multi-armed bandit problem. *SIAM J. Computing*, 32(1):48–77, 2002.

Alina Beygelzimer, John Langford, Lihong Li, Lev Reyzin, and Robert Schapire. Contextual bandit algorithms with supervised learning guarantees. *J. Mach. Learn. Res.*, 15:19–26, 2011.

Róbert Busa-Fekete and Eyke Hüllermeier. A survey of preference-based online learning with bandit algorithms. In *Algorithmic Learning Theory (ALT)*, pages 18–39, 2014.

Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

Olivier Chapelle, Thorsten Joachims, Filip Radlinski, and Yisong Yue. Large-scale validation and analysis of interleaved search evaluation. *ACM Transactions on Information Systems (TOIS)*, 30 (1):6:1–6:41, 2012.

Miroslav Dudík, Daniel Hsu, Satyen Kale, Nikos Karampatziakis, John Langford, Lev Reyzin, and Tong Zhang. Efficient optimal learning for contextual bandits. In *Uncertainty in Artificial Intelligence (UAI)*, pages 169–178, 2011.

Yoav Freund and Robert E. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29:79–103, 1999.

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

Fan Guo, Lei Li, and Christos Faloutsos. Tailoring click models to user goals. In *Workshop on Web Search Click Data (WSCD)*, pages 88–92, 2009a.

Fan Guo, Chao Liu, and Yi-Min Wang. Efficient multiple-click models in web search. In *Proceedings of the International Conference on Web Search and Data Mining (WSDM)*, pages 124–131, 2009b.

Katja Hofmann, Shimon Whiteson, and Maarten de Rijke. A probabilistic method for inferring preferences from clicks. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*, pages 249–258, 2011.

Katja Hofmann, Shimon Whiteson, and Maarten de Rijke. Fidelity, soundness, and efficiency of interleaved comparison methods. *ACM Transactions on Information Systems (TOIS)*, 31(4), 2013.

Adam Kalai and Santosh Vempala. Efficient algorithms for the online decision problem. In *Conference on Learning Theory (COLT)*, pages 26–40, 2003.

John Langford and Tong Zhang. The epoch-greedy algorithm for contextual multi-armed bandits. In *Annual Conference on Neural Information Processing Systems (NIPS)*, pages 817–824, 2007.

Sahand Negahban, Sewoong Oh, and Devavrat Shah. Iterative ranking from pair-wise comparisons. In *Annual Conference on Neural Information Processing Systems (NIPS)*, 2012.

Guillermo Owen. *Game Theory*. Emerald Group Publishing Limited, 3rd edition, 1995.

Filip Radlinski, Madhu Kurup, and Thorsten Joachims. How does clickthrough data reflect retrieval quality? In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*, pages 43–52, 2008.

Markus Schulze. A new monotonic, clone-independent, reversal symmetric, and Condorcet-consistent single-winner election method. *Social Choice and Welfare*, 36(2):267–303, 2011.

Tanguy Urvoy, Fabrice Clerot, Raphael Féraud, and Sami Naamane. Generic exploration and k-armed voting bandits. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 91–99, 2013.

Yisong Yue and Thorsten Joachims. Interactively optimizing information retrieval systems as a dueling bandits problem. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1201–1208, 2009.

Yisong Yue and Thorsten Joachims. Beat the mean bandit. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 241–248, 2011.

Yisong Yue, J. Broder, R. Kleinberg, and T. Joachims. The k-armed dueling bandits problem. In *Conference on Learning Theory (COLT)*, 2009.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 928–936, 2003.

Masrour Zoghi, Shimon Whiteson, Maarten de Rijke, and Rémi Munos. Relative confidence sampling for efficient on-line ranker evaluation. In *Proceedings of the International Conference on Web Search and Data Mining (WSDM)*, pages 73–82, 2014a.

Masrour Zoghi, Shimon Whiteson, Rémi Munos, and Maarten de Rijke. Relative upper confidence bound for the k-armed dueling bandits problem. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 10–18, 2014b.

Masrour Zoghi, Zohar Karnin, Shimon Whiteson, and Maarten de Rijke. Copeland dueling bandits. 2015a. arxiv:1506.00312.

Masrour Zoghi, Shimon Whiteson, and Maarten de Rijke. MergeRUCB: A method for large-scale online ranker evaluation. In *Proceedings of the International Conference on Web Search and Data Mining (WSDM)*, pages 17–26, 2015b.

Appendix A. Failure of the Condorcet winner to exist

Here, we investigate the reliability of the Condorcet assumption by replicating the experiment of (Zoghi et al., 2014a, Section 6.1) with a small modification. As in their setting, we consider a family of K-armed dueling bandit problems arising from the ranker evaluation problem in IR, where the comparisons are carried out using Probabilistic Interleave (Hofmann et al., 2011) and the preferences are generated using click models simulating user behavior (Guo et al., 2009a,b). The rankers are sampled randomly from the set of 136 rankers provided with the MSLR dataset.[6] However, unlike the experiments of Zoghi et al. (2014a), we use an informational click model, rather than a perfect one (Hofmann et al., 2013). The former simulates the behavior of a user who is seeking general information about a broad topic, while the latter represents an idealized user, who meticulously examines every document in the retrieved list. We believe that the informational click model is more realistic and therefore use it here.

The plot in Figure 1 shows the probability with which the encountered dueling bandit problems contain Condorcet winners. As this figure demonstrates, in this setting, the occurrence of the Condorcet winner drops rapidly as the number of rankers grows.

This shows that even in this simple non-contextual example the assumption that there exists a Condorcet winner is too unreliable to be practical. Needless to say that in the contextual dueling bandit problem, where one is dealing with a potentially very large and diverse set of policies, the likelihood of one policy dominating every single other policy is even more unrealistic.

Appendix B. Comparison between the Copeland and von Neumann winners

A Copeland winner is defined to be any arm that beats the largest number of other arms. It is a generalization of the Condorcet winner in the sense that if the Condorcet winner exists, it will be a Copeland winner. However, we claim that the von Neumann winner is a more natural generalization than the Copeland winner for the following two reasons: first, in the absence of a Condorcet winner, Copeland winners, both individually and as a collective, can lose to an arm that is not a Copeland winner, whereas the von Neumann winner beats or ties with every single arm; second, the set of Copeland winners can be altered by the introduction of "clones," i.e., arms whose corresponding rows of the preference matrix are identical to each other.

To demonstrate this lack of stability of Copeland winners, consider any $K + 3$-armed example with $K > 4$, where arms a_1, a_2 and a_3 beat all other arms and the three of them are in a cycle, with a_1 beating a_2, a_2 beating a_3 and a_3 beating a_1 all with probability 1. It is easy to see that these three arms are the only Copeland winners with Copeland score equal to $K + 1$ and also form the support of the von Neumann distribution: indeed, the von Neumann distribution is simply the uniform distribution on these three arms. Now, let us consider a slight modification of this problem, where we add one more arm, called a_0, which is a duplicate of arm a_1; hence $P(0, 1) = 0$ and $P(0, j) = P(1, j)$ for all $j > 1$. In the following we explain what happens to the set of Copeland winners after this modification. In the presence of ties, there are three sensible definitions that one could use for the Copeland score; these definitions and the corresponding scores for the top four arms in our modified example can be found in Table 1. As the quantities in this table show, regardless of the definition of the Copeland score used, the set of Copeland winners for our new

---

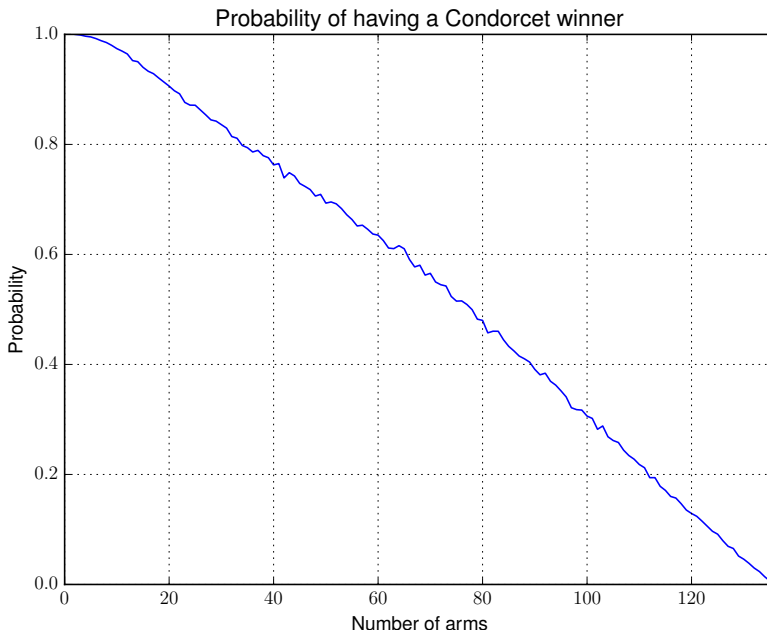6. http://research.microsoft.com/en-us/projects/mslr/default.aspx

Figure 1: The probability that the Condorcet assumption holds for subsets of the feature rankers in the MSLR dataset. The probability is shown as a function of the size of the subset of rankers under consideration.

Table 1: Copeland scores of the top arms in the duplicated example

The Copeland score variant (for a_i)	$\|\{j : P(i,j) > 0\}\|$	$\|\{j : P(i,j) \geq 0\}\|$	$\|\{j : P(i,j) > 0\}\|$ $- \|\{j : P(i,j) < 0\}\|$
$i = 0, 1$	$K + 1$	$K + 3$	K
$i = 2$	$K + 1$	$K + 2$	$K - 1$
$i = 3$	$K + 2$	$K + 3$	$K + 1$

$K + 4$-armed dueling bandit problem does not contain all of $a_0, \ldots, a_3$. Indeed, under no definition can arm a_2 be considered a Copeland winner.

On the other hand, arms a_0, a_1, a_2, a_3 still form the support of the von Neumann distribution of this modified dueling bandit problem: if we assign weights $w(0), w(1), w(2), w(3)$ to these four arms such that

$$w(0) + w(1) = w(2) = w(3) = \frac{1}{3}$$

and sample an arm a_i according to these weights, we will (on average) beat any a_j with $j > 3$ and tie with all a_j with $j \leq 3$.

We consider the lack of stability under cloning illustrated by this example to be a major drawback of the Copeland score as a measure of quality.
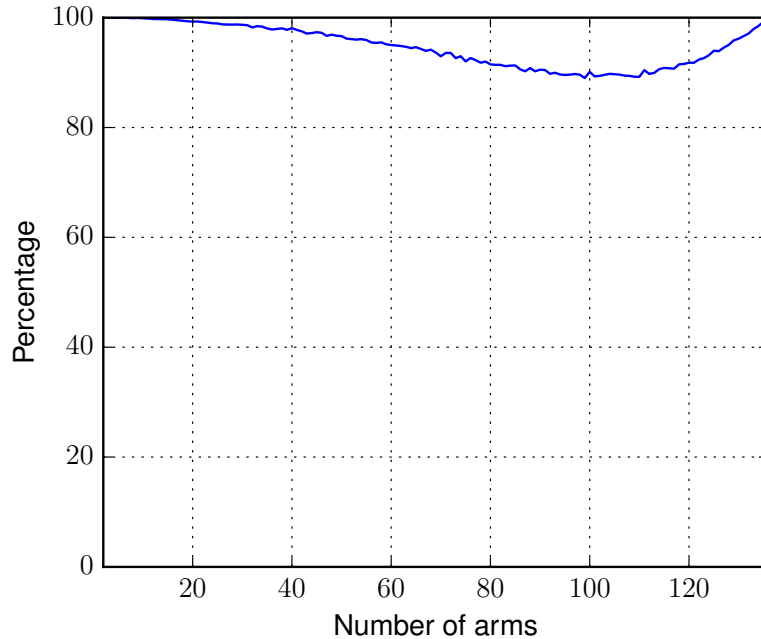
Figure 2: The percentage of preference matrices of a given size, sampled from the MSLR dataset, for which all Copeland winners are contained in the support of the von Neumann winner

Furthermore, as the following 5-armed preference matrix illustrates, the von Neumann winner does not necessarily contain the Copeland winner in its support:

$$
\mathbf{P} = \begin{bmatrix}
0 & 0.5 & -0.5 & 0.5 & -0.95 \\
-0.5 & 0 & 0.5 & -0.2 & 0.5 \\
0.5 & -0.5 & 0 & -0.2 & 0.5 \\
-0.5 & 0.2 & 0.2 & 0 & 0.5 \\
0.95 & -0.5 & -0.5 & -0.5 & 0
\end{bmatrix}
$$

Indeed, the von Neumann winner of this matrix is the uniform distribution on the first three arms, as can be easily checked by multiplying the row vector $[1/3 \ 1/3 \ 1/3 \ 0 \ 0]$ with $\mathbf{P}$, while the Copeland winner is the fourth arm, since the Copeland scores of the 5 arms are $[2 \ 2 \ 2 \ 3 \ 1]$. Moreover, the fourth arm also happens to be both the Borda winner (Urvoy et al., 2013) and the Random Walk winner (Negahban et al., 2012; Busa-Fekete and Hüllermeier, 2014). The Borda winner is the arm with the highest chance of winning a comparison against a uniformly randomly chosen opponent, i.e., the arm corresponding to the row in the preference matrix whose entries have the highest sum: in this case the Borda scores are $[0.455 \ 0.53 \ 0.53 \ 0.54 \ 0.445]$. The Random Walk winner is obtained as follows: first, we convert the "probabilistic" preference matrix (i.e., $\mathbf{P}/2 + 0.5$) into a column-wise stochastic matrix (by dividing each column by its sum), then find the stationary distribution of the Markov chain defined by this matrix (by finding the right eigenvector of the stochastic matrix corresponding to eigenvalue 1), and, finally, declare the arm with the highest probability under this
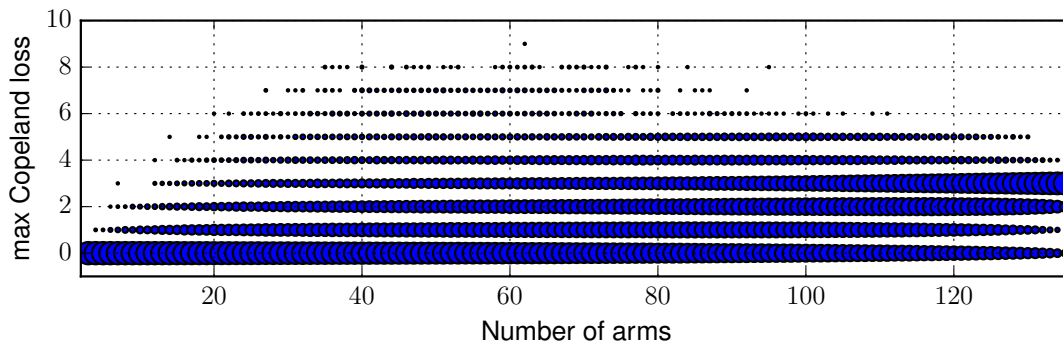
Figure 3: The fractions of preference matrices of a given size (horizontal axis) with a given maximum Copeland loss among the arms in the support of their von Neumann winner (vertical axis): given K and M, the area of the circle at coordinate (K, M) of this plot is proportional to the percentage of K-armed sub-matrices of the Informational MSLR preference matrix, $\mathbf{P}$, for which the maximum Copeland loss of an arm in the support of the von Neumann winner of $\mathbf{P}$ is equal to M.

stationary distribution to be the Random Walk winner. In this particular example, the stationary distribution is $[0.198 \ 0.212 \ 0.204 \ 0.217 \ 0.169]$ and so the Random Walk winner is the fourth arm, as mentioned before.

Despite the above observations, in practice, the Copeland winner and the von Neumann winner tend to agree to a large extent. For instance, in preference matrices sampled from the MSLR dataset, as described in Appendix A, in over 99.9% of the examples, the von Neumann winner contained at least one Copeland winner. Moreover, in the overwhelming majority of the cases, all Copeland winners were assigned non-zero probability by the von Neumann winner, although the percentage of cases where this phenomenon occurs is slightly lower than the above figure and dependent on the number of arms (see Figure 2). Based on these observations, the Copeland winner can roughly be thought of as a more restrictive notion than the von Neumann winner.

Furthermore, as Figure 3 demonstrates, the arms in the support of the von Neumann winner tend to have high Copeland scores (or equivalently, low Copeland losses) in practice. Given this close relation between these two notions of winners, a natural question becomes whether the recent improvements made in solving the Copeland dueling bandit problem (Zoghi et al., 2015a) can be used to speed up the task of finding the von Neumann winner.

Another aspect of the von Neumann winner that might be disconcerting when first encountered is the fact that it is a distribution, which in theory can put non-zero probability on all arms; however, in practice, this is very far from being the case. Indeed, among the over a million preference matrices sampled from the MSLR dataset, not a single one had a von Neumann winner that assigned non-zero probability to more than 5 arms. In fact, in the vast majority of the cases, the von Neumann winners had supports of size 1 or 3 (see Figure 4). Note that fewer than 0.03% of preference matrices had von Neumann winners whose support contains 5 arms.
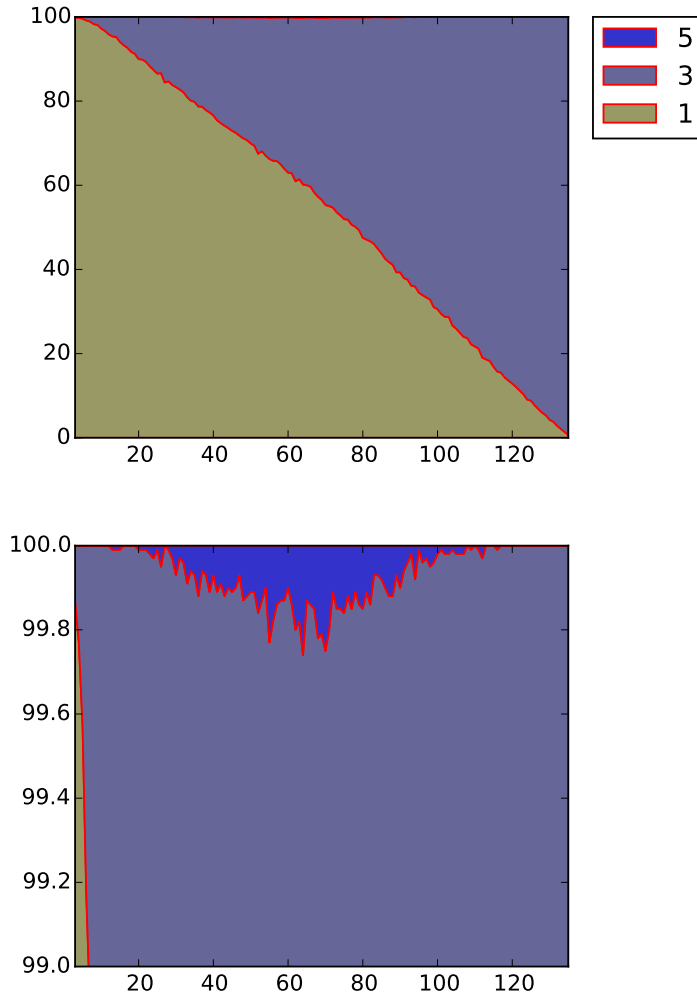
Figure 4: The percentages of preference matrices of a given size with von Neumann winners of a given support size: the bottom plot is simply a zoomed version of the top one, and it was included, because preference matrices with von Neumann winners of support size 5 were very infrequent.

Appendix C. Analysis of `SparringEXP4.P` (proof of Theorem 2)

To fulfill the requirements of the learning model for `Exp4.P`, we also need to define rewards $r_t(a)$ for all of the actions that were *not* chosen. Furthermore, these rewards need to be defined *before* each copy of the algorithm chooses its action (or, more technically, in a manner that is conditionally independent of each copy's choice). To this end, for every pair of actions (a, b), we define a $\{-1, +1\}$-valued random variable $R_t(a, b)$ with the expected value $P_t(a, b)$. Thus, $R_t(a, b)$ can be viewed as the outcome of a hypothetical duel between actions a and b. These values are only used

for the mathematical argument, and do not literally have to be computed. Only the pair (a_t, b_t) is actually used in a duel.

For row-Exp, based on column-Exp's chosen action b_t, we then define rewards $r_t^{(row)}(a) = R_t(a, b_t)$ for all a. And similarly, for column-Exp, based on row-Exp's chosen action a_t, we define rewards $r_t^{(col)}(b) = -R_t(a_t, b)$ for all b. In particular, this means that row-Exp receives, for its chosen action a_t, the reward $R_t(a_t, b_t)$ (that is, the result of a duel between a_t and b_t), while column-Exp receives the reward $-R_t(a_t, b_t)$ for its chosen action b_t.

Let us first take the point of view of row-Exp. Plugging in to Eq. (4), we have that, with probability at least $1 - \delta/4$, for all $\pi \in \Pi$,

$$\sum_{t=1}^{T} R_t(a_t, b_t) \geq \sum_{t=1}^{T} R_t(\pi(x_t), b_t) - O(\sqrt{KT \ln(|\Pi|/\delta)}).$$

Further, using Azuma's lemma and union bound, we can show that, with probability at least $1 - \delta/4$, for every $\pi \in \Pi$

$$\sum_{t=1}^{T} R_t(\pi(x_t), b_t) \geq \sum_{t=1}^{T} P_t(\pi(x_t), b_t) - O(\sqrt{KT \ln(|\Pi|/\delta)}).$$

Similarly, from column-Exp's perspective, with probability at least $1 - \delta/4$, for all $\pi \in \Pi$,

$$-\sum_{t=1}^{T} R_t(a_t, b_t) \geq -\sum_{t=1}^{T} R_t(a_t, \pi(x_t)) - O(\sqrt{KT \ln(|\Pi|/\delta)})$$

and, by Azuma's lemma and the skew-symmetry of $\mathbf{P}_t$, with probability at least $1 - \delta/4$, for every $\pi \in \Pi$,

$$-\sum_{t=1}^{T} R_t(a_t, \pi(x_t)) \geq \sum_{t=1}^{T} P_t(\pi(x_t), a_t) - O(\sqrt{KT \ln(|\Pi|/\delta)}).$$

Combining and rearranging now yields the theorem.

Appendix D. Analysis of `SparringFPL` and `ProjectedGD`

Compared to the presentation in the body of the paper, we extend our setting from uniform exploration strategy to an arbitrary unbiased estimator $\hat{\mathbf{P}}_i$ of $\mathbf{P}_i$, i.e., to any matrix $\hat{\mathbf{P}}_i$ which satisfies $\mathbb{E}[\hat{\mathbf{P}}_i \mid \mathbf{P}_i] = \mathbf{P}_i$. Our results are parameterized by two numbers, L, V, such that $|\hat{P}_i(a, b)| \leq L$ and $\text{Var}(\hat{P}_i(a, b)) \leq V$ for all exploration rounds i and all action pairs (a, b). For uniform exploration, both upper bounds are K^2.

D.1. Reduction from M to M̂ (proof of Lemma 3)

In this subsection, we prove Lemma 3 which reduces the optimization problem to that on the approximate matrix $\hat{\mathbf{M}}$ computed from the data. As a first step, we prove Eq. (6) which relates $\hat{\mathbf{M}}$ to the true preference matrix for $\mathbf{M}$.

Proof [Eq. (6)] Let $Z_1, \ldots, Z_m$ be independent, identically distributed random variables, each taking values in $[-R, R]$, and each having mean zero and variance V. Then according to Bernstein's inequality, the probability that the average $A = (1/m) \sum_{i=1}^{m} Z_i$ exceeds some value s is at most

$$\exp\left(-\frac{s^2/2}{mV + Rs/3}\right).$$

For an appropriate choice of s, this implies that, with probability at least $1 - \delta$,

$$A \leq \frac{2R \ln(1/\delta)}{3m} + \sqrt{\frac{2V \ln(1/\delta)}{m}}. \tag{11}$$

To derive Eq. (6), for a fixed pair of policies π and ρ, we can let $Z_i = \hat{P}_i(\pi(x_i), \rho(x_i)) - M(\pi, \rho)$ whose mean is zero and variance is at most V; further, $|Z_i| \leq 1 + L$. Plugging into Eq. (11) implies that Eq. (6) holds with

$$\varepsilon' = \frac{2(1 + L) \ln(|\Pi|^2/\delta)}{3m} + \sqrt{\frac{2V \ln(|\Pi|^2/\delta)}{m}}$$

with probability at least $1 - \delta/|\Pi|^2$. By the union bound, with probability at least $1 - \delta$, this will hold simultaneously for *all* policies π and ρ. ∎

Proof [Lemma 3] Eq. (6) implies that $\mathbf{W}^\top \hat{\mathbf{M}} \mathbf{U}$ is within ε' of $\mathbf{W}^\top \mathbf{M} \mathbf{U}$, for all probability vectors $\mathbf{W}$ and $\mathbf{U}$. Therefore,

$$
\begin{aligned}
\min_{\mathbf{U} \in \Delta_{|\Pi|}} \hat{\mathbf{W}}^\top \mathbf{M} \mathbf{U} \;\; &\geq \;\; \min_{\mathbf{U} \in \Delta_{|\Pi|}} \hat{\mathbf{W}}^\top \hat{\mathbf{M}} \mathbf{U} - \varepsilon' \\
&\geq \;\; \max_{\mathbf{W} \in \Delta_{|\Pi|}} \min_{\mathbf{U} \in \Delta_{|\Pi|}} \mathbf{W}^\top \hat{\mathbf{M}} \mathbf{U} - \varepsilon' - \varepsilon \\
&\geq \;\; \max_{\mathbf{W} \in \Delta_{|\Pi|}} \min_{\mathbf{U} \in \Delta_{|\Pi|}} \mathbf{W}^\top \mathbf{M} \mathbf{U} - 2\varepsilon' - \varepsilon \\
&= \;\; -(2\varepsilon' + \varepsilon).
\end{aligned}
$$

∎

D.2. Analysis of `SparringFPL`

To analyze `SparringFPL`, we build on the provable guarantees for FPL.

For convenience, let us recap the learning setting for FPL. Let $\mathcal{D}$ and $\mathcal{L}$ be subsets of $\mathbb{R}^{mK}$. On each round $t = 1, \ldots, N$, the learner chooses a decision vector $\mathbf{d}_t \in \mathcal{D}$, and then receives a loss vector $\boldsymbol{\ell}_t \in \mathcal{L}$. The learner's goal is to minimize its cumulative loss $\sum_{t=1}^{N} \mathbf{d}_t \cdot \boldsymbol{\ell}_t$ relative to the best possible loss using a fixed decision, that is, $\min_{\mathbf{d} \in \mathcal{D}} \sum_{t=1}^{N} \mathbf{d} \cdot \boldsymbol{\ell}_t$.

Kalai and Vempala (2003, Theorem 1.1) prove the following (slightly simplified) result: assume that D, R and A are such that for all $\mathbf{d} \in \mathcal{D}$ and $\boldsymbol{\ell} \in \mathcal{L}$ we have that $\|\mathbf{d}\|_1 \leq D$, $|\mathbf{d} \cdot \boldsymbol{\ell}| \leq R$ and $\|\boldsymbol{\ell}\|_1 \leq A$. Also, let $\alpha = \sqrt{2D/(RAN)}$. Then, for any sequence $\boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_N \in \mathcal{L}$,

$$\mathbb{E}\left[\sum_{t=1}^{N} \mathbf{d}_t \cdot \boldsymbol{\ell}_t\right] \leq \min_{\mathbf{d} \in \mathcal{D}} \sum_{t=1}^{N} \mathbf{d} \cdot \boldsymbol{\ell}_t + 2\sqrt{2DRAN}. \tag{12}$$

where the expectation is over the random choice of perturbations. Kalai and Vempala prove this in the oblivious case when the adversary has fixed the ℓ_t's ahead of time. However, this restriction can be relaxed to allow each ℓ_t to be selected adaptively in a possibly stochastic fashion that may depend on the entire preceding history through round $t - 1$, but *not* on the perturbation $\mathbf{p}_t$ for the current round. Using a martingale argument and Azuma's lemma (see also Cesa-Bianchi and Lugosi, 2006, Lemma 4.1), it can then be shown that, with probability at least $1 - \delta'$,

$$\frac{1}{N}\sum_{t=1}^{N}\mathbf{d}_t \cdot \ell_t \leq \min_{\mathbf{d}\in\mathcal{D}}\frac{1}{N}\sum_{t=1}^{N}\mathbf{d}\cdot\ell_t + 2\sqrt{2DRA/N} + 2R\sqrt{2\ln(1/\delta')/N}. \tag{13}$$

Theorem 6 *In* `SparringFPL`*, set parameter* $\alpha = \sqrt{2/(L^2N)}$*. Then with probability at least* $1-\delta$*, the vector* $\overline{\mathbf{w}}$ *returned by the algorithm satisfies*

$$\min_{\mathbf{u}\in\mathcal{C}}\overline{\mathbf{w}}^\top\mathbf{B}\mathbf{u} \geq \max_{\mathbf{w}\in\mathcal{C}}\min_{\mathbf{u}\in\mathcal{C}}\mathbf{w}^\top\mathbf{B}\mathbf{u} - 2\varepsilon$$

where $\varepsilon = 2L\sqrt{2m/N} + 2L\sqrt{2\ln(2/\delta)/N}$*.*

Thus, to find an ε-approximate solution, we can choose N to be $O(L^2/\varepsilon^2)(m + \ln(1/\delta))$. This also gives a bound on the number of oracle calls (it is called twice per round).

Proof Note that in our case, we can choose $D = \sqrt{m}$, $A = L\sqrt{m}$ and $R = L$.

Let $\overline{\mathbf{u}} = \frac{1}{N}\sum_{t=1}^{N}\mathbf{u}_t$. Then we have the following chain of inequalities holding with probability at least $1 - \delta$:

$$\min_{\mathbf{u}\in\mathcal{C}}\max_{\mathbf{w}\in\mathcal{C}}\mathbf{w}^\top\mathbf{B}\mathbf{u} - \varepsilon \quad \leq \quad \max_{\mathbf{w}\in\mathcal{C}}\mathbf{w}^\top\mathbf{B}\overline{\mathbf{u}} - \varepsilon$$

$$\leq \quad \frac{1}{N}\sum_{t=1}^{N}\mathbf{w}_t^\top\mathbf{B}\mathbf{u}_t \tag{14}$$

$$\leq \quad \min_{\mathbf{u}\in\mathcal{C}}\overline{\mathbf{w}}^\top\mathbf{B}\mathbf{u} + \varepsilon \tag{15}$$

$$\leq \quad \max_{\mathbf{w}\in\mathcal{C}}\min_{\mathbf{u}\in\mathcal{C}}\mathbf{w}^\top\mathbf{B}\mathbf{u} + \varepsilon.$$

Here, Eqs. (14) and (15) follow directly from Eq. (13) applied, respectively, to row-FPL and column-FPL with $\delta' = \delta/2$. Noting that

$$\max_{\mathbf{w}\in\mathcal{C}}\min_{\mathbf{u}\in\mathcal{C}}\mathbf{w}^\top\mathbf{B}\mathbf{u} = \min_{\mathbf{u}\in\mathcal{C}}\max_{\mathbf{w}\in\mathcal{C}}\mathbf{w}^\top\mathbf{B}\mathbf{u},$$

the theorem now follows. ∎

D.3. Analysis of `ProjectedGD`: outer loop

Using an analysis similar to Zinkevich (2003), but for a fixed learning rate, and taking into account the errors introduced by imperfect projections, we can show the following:

Lemma 7 *For the algorithm* `ProjectedGD` *with* $\eta = 2/(L\sqrt{N_{out}})$, *we have*

$$\frac{1}{N_{out}} \sum_{t=1}^{N_{out}} \mathbf{w}_t^\top \mathbf{B} \mathbf{u}_t \geq \max_{\mathbf{w} \in \mathcal{C}} \frac{1}{N_{out}} \sum_{t=1}^{N_{out}} \mathbf{w}^\top \mathbf{B} \mathbf{u}_t - \varepsilon$$

where $\varepsilon = 2L/\sqrt{N_{out}} + L\alpha/2$.

Proof For all $\mathbf{w} \in \mathcal{C}$, we have

$$
\begin{aligned}
\|\mathbf{w} - \mathbf{w}_{t+1}\|^2 - \|\mathbf{w} - \mathbf{w}_t\|^2 &\leq& \|\mathbf{w} - \mathbf{z}_{t+1}\|^2 - \|\mathbf{w} - \mathbf{w}_t\|^2 + \alpha\eta L & (16) \\
&=& -2\eta(\mathbf{w} - \mathbf{w}_t)^\top \mathbf{B} \mathbf{u}_t + \eta^2 \|\mathbf{B}\mathbf{u}_t\|^2 + \alpha\eta L & (17) \\
&\leq& -2\eta(\mathbf{w} - \mathbf{w}_t)^\top \mathbf{B} \mathbf{u}_t + \eta^2 L^2 + \alpha\eta L.
\end{aligned}
$$

Here, Eq. (16) uses Eq. (9), applied to our case where we have $\mathbf{z}_{t+1} - \mathbf{w}_t = \eta \mathbf{B} \mathbf{u}_t$. Eq. (17) follows from straightforward algebra. Since $\|\mathbf{w} - \mathbf{w}_1\| \leq 2$, summing over $t = 1, \ldots, N_{out}$ yields, for all $\mathbf{w} \in \mathcal{C}$,

$$
\begin{aligned}
-4 &\leq& \|\mathbf{w} - \mathbf{w}_{N_{out}+1}\|^2 - \|\mathbf{w} - \mathbf{w}_1\|^2 \\
&\leq& -2\eta \sum_{t=1}^{N_{out}} \mathbf{w}^\top \mathbf{B} \mathbf{u}_t + 2\eta \sum_{t=1}^{N_{out}} \mathbf{w}_t^\top \mathbf{B} \mathbf{u}_t + \eta^2 L^2 N_{out} + \alpha\eta L N_{out}.
\end{aligned}
$$

Re-arranging completes the lemma. ■

We can prove that the returned vector $\overline{\mathbf{w}}$ is an ε-approximate maxmin solution using a technique similar to Freund and Schapire (1999). (Alternatively, we could use the average of the $\mathbf{u}_t$'s which is an ε-approximate minmax solution by the same proof.)

Theorem 8 *The vector* $\overline{\mathbf{w}}$ *satisfies* $\min_{\mathbf{u} \in \mathcal{C}} \overline{\mathbf{w}}^\top \mathbf{B} \mathbf{u} \geq \max_{\mathbf{w} \in \mathcal{C}} \min_{\mathbf{u} \in \mathcal{C}} \mathbf{w}^\top \mathbf{B} \mathbf{u} - \varepsilon$ *where* ε *is as Lemma 7.*

Proof Let $\overline{\mathbf{u}} = \frac{1}{N_{out}} \sum_{t=1}^{N_{out}} \mathbf{u}_t$. Then

$$
\begin{aligned}
\max_{\mathbf{w} \in \mathcal{C}} \min_{\mathbf{u} \in \mathcal{C}} \mathbf{w}^\top \mathbf{B} \mathbf{u} &\geq& \min_{\mathbf{u} \in \mathcal{C}} \overline{\mathbf{w}}^\top \mathbf{B} \mathbf{u} \\
&=& \min_{\mathbf{u} \in \mathcal{C}} \frac{1}{N_{out}} \sum_{t=1}^{N_{out}} \mathbf{w}_t^\top \mathbf{B} \mathbf{u} \\
&\geq& \frac{1}{N_{out}} \sum_{t=1}^{N_{out}} \min_{\mathbf{u} \in \mathcal{C}} \mathbf{w}_t^\top \mathbf{B} \mathbf{u} \\
&=& \frac{1}{N_{out}} \sum_{t=1}^{N_{out}} \mathbf{w}_t^\top \mathbf{B} \mathbf{u}_t \\
&\geq& \max_{\mathbf{w} \in \mathcal{C}} \frac{1}{N_{out}} \sum_{t=1}^{N_{out}} \mathbf{w}^\top \mathbf{B} \mathbf{u}_t - \varepsilon \\
&=& \max_{\mathbf{w} \in \mathcal{C}} \mathbf{w}^\top \mathbf{B} \overline{\mathbf{u}} - \varepsilon \\
&\geq& \min_{\mathbf{u} \in \mathcal{C}} \max_{\mathbf{w} \in \mathcal{C}} \mathbf{w}^\top \mathbf{B} \mathbf{u} - \varepsilon.
\end{aligned}
$$

■

D.4. Analysis of `ProjectedGD`: inner loop

It remains to analyze the inner loop of `ProjectedGD`, i.e., the approximate-projection procedure `ApproxProject(z, v₁)`.

Let $\mathbf{v}^*$ be the projection of $\mathbf{z}$ onto the policy hull $\mathcal{C}$. We can prove the following for this algorithm using $\|\mathbf{v}^* - \mathbf{v}_t\|^2$ as a potential function.

Lemma 9 *For the algorithm* `ApproxProject` *with* $\nu = \|\mathbf{z} - \mathbf{v}_1\|/\sqrt{N_{in}}$ *and* $\delta = 8\|\mathbf{z} - \mathbf{v}_1\|/\sqrt{N_{in}}$, *we have*

$$\frac{1}{N_{in}} \sum_{t=1}^{N_{in}} F(\mathbf{s}_t, \mathbf{v}_t) \geq \frac{1}{N_{in}} \sum_{t=1}^{N_{in}} F(\mathbf{s}_t, \mathbf{v}^*) - \delta.$$

Proof We have

$$
\begin{aligned}
\|\mathbf{v}^* - \mathbf{v}_{t+1}\|^2 - \|\mathbf{v}^* - \mathbf{v}_t\|^2 &= \|\mathbf{v}^* - \mathbf{v}_t + \nu(\mathbf{v}_t - \mathbf{s}_t)\|^2 - \|\mathbf{v}^* - \mathbf{v}_t\|^2 \\
&= 2\nu(\mathbf{v}^* - \mathbf{v}_t) \cdot (\mathbf{v}_t - \mathbf{s}_t) + \nu^2 \|\mathbf{v}_t - \mathbf{s}_t\|^2 \\
&\leq 2\nu(\mathbf{v}^* - \mathbf{v}_t) \cdot (\mathbf{v}_t - \mathbf{s}_t) + 4\nu^2 \qquad (18) \\
&\leq \nu(F(\mathbf{s}_t, \mathbf{v}_t) - F(\mathbf{s}_t, \mathbf{v}^*)) + 4\nu^2. \qquad (19)
\end{aligned}
$$

Eq. (18) uses $\|\mathbf{v}_t - \mathbf{s}_t\| \leq \|\mathbf{v}_t\| + \|\mathbf{s}_t\| \leq 2$. To see Eq. (19), note that

$$
\begin{aligned}
2(\mathbf{v}^* - \mathbf{v}_t) \cdot (\mathbf{v}_t - \mathbf{s}_t) &= 2\mathbf{v}^* \cdot \mathbf{v}_t - 2\|\mathbf{v}_t\|^2 - 2\mathbf{s}_t \cdot (\mathbf{v}^* - \mathbf{v}_t) \\
&\leq \|\mathbf{v}^*\|^2 + \|\mathbf{v}_t\|^2 - 2\|\mathbf{v}_t\|^2 - 2\mathbf{s}_t \cdot (\mathbf{v}^* - \mathbf{v}_t) \\
&= \left[2\mathbf{s}_t \cdot (\mathbf{v}_t - \mathbf{z}) + \|\mathbf{z}\|^2 - \|\mathbf{v}_t\|^2 \right] \\
&\quad - \left[2\mathbf{s}_t \cdot (\mathbf{v}^* - \mathbf{z}) + \|\mathbf{z}\|^2 - \|\mathbf{v}^*\|^2 \right] \\
&= F(\mathbf{s}_t, \mathbf{v}_t) - F(\mathbf{s}_t, \mathbf{v}^*).
\end{aligned}
$$

The inequality here uses the fact that, for any two vectors $\mathbf{u}$ and $\mathbf{w}$, we have $2\mathbf{u} \cdot \mathbf{w} \leq \|\mathbf{u}\|^2 + \|\mathbf{w}\|^2$.

Also,

$$
\begin{aligned}
\|\mathbf{v}^* - \mathbf{v}_1\| &\leq \|\mathbf{v}^* - \mathbf{z}\| + \|\mathbf{z} - \mathbf{v}_1\| \\
&= \min_{\mathbf{v} \in \mathcal{C}} \|\mathbf{v} - \mathbf{z}\| + \|\mathbf{z} - \mathbf{v}_1\| \\
&\leq \|\mathbf{v}_1 - \mathbf{z}\| + \|\mathbf{z} - \mathbf{v}_1\|. \qquad (20)
\end{aligned}
$$

Summing Eq. (19) for $t = 1, \ldots, N_{in}$ and combining with Eq. (20) gives

$$
\begin{aligned}
-4\|\mathbf{z} - \mathbf{v}_1\|^2 &\leq \|\mathbf{v}^* - \mathbf{v}_{N_{in}+1}\|^2 - \|\mathbf{v}^* - \mathbf{v}_1\|^2 \\
&\leq \nu \sum_{t=1}^{N_{in}} F(\mathbf{s}_t, \mathbf{v}_t) - \nu \sum_{t=1}^{N_{in}} F(\mathbf{s}_t, \mathbf{v}^*) + 4\nu^2 N_{in}.
\end{aligned}
$$

Re-arranging and applying our choice of ν completes the lemma. ∎

Next, we show that $\overline{\mathbf{v}}$ satisfies the specification given in Eq. (10).

Theorem 10 *For the algorithm* `ApproxProject`, $\min_{\mathbf{s} \in \mathcal{C}} F(\mathbf{s}, \overline{\mathbf{v}}) \geq -\delta$ *with* δ *set as in Lemma 9. Thus, Eq. (10) holds for* $\overline{\mathbf{v}}$ *if we set* $\alpha = 8/\sqrt{N_{in}}$.

Proof Let

$$\overline{\mathbf{s}} = \frac{1}{N_{in}} \sum_{t=1}^{N_{in}} \mathbf{s}_t.$$

Then

$$
\begin{aligned}
\min_{\mathbf{s} \in \mathcal{C}} F(\mathbf{s}, \overline{\mathbf{v}}) \ &\geq\ \min_{\mathbf{s} \in \mathcal{C}} \frac{1}{N_{in}} \sum_{t=1}^{N_{in}} F(\mathbf{s}, \mathbf{v}_t) && (21) \\
&\geq\ \frac{1}{N_{in}} \sum_{t=1}^{N_{in}} \min_{\mathbf{s} \in \mathcal{C}} F(\mathbf{s}, \mathbf{v}_t) && \\
&=\ \frac{1}{N_{in}} \sum_{t=1}^{N_{in}} F(\mathbf{s}_t, \mathbf{v}_t) && (22) \\
&\geq\ \frac{1}{N_{in}} \sum_{t=1}^{N_{in}} F(\mathbf{s}_t, \mathbf{v}^*) - \delta && \\
&=\ F(\overline{\mathbf{s}}, \mathbf{v}^*) - \delta && (23) \\
&\geq\ \min_{\mathbf{s} \in \mathcal{C}} F(\mathbf{s}, \mathbf{v}^*) - \delta && \\
&\geq\ \|\mathbf{z} - \mathbf{v}^*\|^2 - \delta \geq -\delta. && (24)
\end{aligned}
$$

Eq. (21) uses Jensen's inequality and the fact that $F(\mathbf{s}, \cdot)$ is concave for each $\mathbf{s}$. Eq. (22) follows from our choice of $\mathbf{s}_t$ (which minimizes $F(\cdot, \mathbf{v}_t)$). Eq. (23) uses linearity of $F(\cdot, \mathbf{v})$ for each $\mathbf{v}$. And Eq. (24) uses $F(\mathbf{s}, \mathbf{v}^*) \geq \|\mathbf{z} - \mathbf{v}^*\|^2$ for all $\mathbf{s} \in \mathcal{C}$, which follows from simple Euclidean geometry and the Pythagorean theorem. ■

Finally, combining with Lemma 7 and Theorem 8, this shows that the overall solution $\overline{\mathbf{w}}$ will be an ε-approximate maxmin solution where $\varepsilon = 2L/\sqrt{N_{out}} + 4L/\sqrt{N_{in}}$. Thus, we can obtain any desired value of ε by setting $N_{in} = N_{out} = \lceil 36L^2/\varepsilon^2 \rceil$. The resulting number of calls to the classification oracle will be $N_{out} + N_{in}N_{out} = O(L^4/\varepsilon^4)$. As earlier noted, compared to `SparringFPL`, this bound gives a different trade-off between m and ε. For the case that $\varepsilon = O(\varepsilon')$, and with ε' as in Section 6, this algorithm gives a better bound by a factor of $O((\ln|\Pi|)/K^2)$.