Bandits with Abstention under Expert Advice

Anonymous Author(s)
Affiliation
Address
email

Abstract

We study the classic problem of prediction with expert advice under bandit feedback. Our model assumes that one action, corresponding to the learner's abstention from play, has no reward or loss on every trial. We propose the *confidence-rated bandits with abstentions* (CBA) algorithm, which exploits this assumption to obtain reward bounds that can significantly improve those of the classical EXP4 algorithm. Our problem can be construed as the aggregation of confidence-rated predictors, with the learner having the option to abstain from play. We are the first to achieve bounds on the expected cumulative reward for general confidence-rated predictors. In the special case of specialists we achieve a novel reward bound, significantly improving previous bounds of SPECIALISTEXP (treating abstention as another action). We discuss how CBA can be applied to the problem of adversarial contextual bandits with the option of abstaining from selecting any action. We are able to leverage a wide range of inductive biases, outperforming previous approaches both theoretically and in preliminary experimental analysis. Additionally, we achieve a reduction in runtime from quadratic to almost linear in the number of contexts for the specific case of metric space contexts.

1 Introduction

We study the classic problem of prediction with expert advice under bandit feedback. The problem is structured as a sequence of trials. During each trial, each expert recommends a probability distribution over the set of possible actions. The learner then selects an action, observes, and incurs the (potentially negative) reward associated with that action on that particular trial. In practical applications, errors often lead to severe consequences, and consistently making predictions is neither safe nor economically practical. For this reason, the abstention option has gained a lot of interest in the literature, both in the batch and online setting [Chow, 1957, 1970, Hendrickx et al., 2021, Cortes et al., 2018]. Similarly to previous works, this paper is based on the assumption that one of the actions always has zero reward: such an action is equivalent to an abstention of the learner from play. Besides the rewards being bounded between $[-1, 1]$, we make no additional assumptions regarding how the rewards or expert predictions are generated. In this paper, we present an efficient algorithm CBA (**C**onfidence-rated **B**andits with **A**bstentions) which exploits the abstention action to get reward bounds that can be dramatically higher than those of EXP4 [Auer et al., 2002]. In the worst case, our reward bound essentially matches that of EXP4 so that CBA can be seen as a strict improvement.

Our problem can also be seen as that of aggregating *confidence-rated predictors* [Blum and Mansour, 2007, Gaillard et al., 2014, Luo and Schapire, 2015] when the learner has the option of abstaining from taking actions. When the problem is phrased in this way, at the start of each trial, each predictor recommends a probability distribution over the actions (which now may not include an action with zero reward) but with a confidence rating. A low confidence rating can mean that either the predictor thinks that all actions are bad (so that the learner should abstain) or simply does not know which

Submitted to the ICML 2024 Workshop on Foundations of Reinforcement Learning and Control. Do not distribute.

action is the best. Previous works on confidence-rated experts measure the performance of their algorithm in terms of the sum of *scaled* per-trial rewards. In contrast to previous algorithms, our approach allows for the derivation of bounds on the expected cumulative reward of CBA.

This formulation enables us to extend our work to the problem of adversarial contextual bandits with the abstention option, which has not been studied before. Previous work has considered the abstention option in the standard (context-free) adversarial bandit setting or in stochastic settings [Cortes et al., 2018, 2020, Neu and Zhivotovskiy, 2020], but not in the contextual and adversarial case. Moreover, their results and methods cannot be applied to confidence-rated predictors. To get more intuition on this setup, we can think of any deterministic policy that maps contexts into actions. Any such policy can be viewed as a classifier, with *foreground* classes associated with each action and a *background* class associated with abstaining. Our learning bias is represented by a set of information we refer to as the *basis*, which we formally define later. It encodes contextual structural assumptions that hold exclusively for the foreground classes and are provided to the algorithm a priori. A particular type of basis is generated by a set of potential clusters that can overlap. Alternatively, a basis can also be created using balls generated by any kind of distance function, which groups contexts believed to be close together. For this latter family of basis, we can also achieve a significant speedup in the per-trial time complexity of CBA.

One specific scenario where prior algorithms can establish cumulative reward bounds is as follows: on any given trial, the predictors are *specialists* [Freund et al., 1997], having either full confidence (a.k.a. *awake*) or no confidence (a.k.a. *asleep*). The SPECIALISTEXP algorithm by Herbster et al. [2021], a bandit version of the standard specialist algorithm, achieves regret bounds with respect to any subset of specialists where exactly one specialist is awake on each trial. We differ from this work as abstention is an algorithmic choice. Instead of sleeping in the rounds where the specialist is not active, the specialist may vote for abstention, which is a proper action of our algorithm. In Section 5.2, we present an illustrative problem involving learning balls in a space equipped with a metric. This example demonstrates our capability to significantly improve SPECIALISTEXP, which treats abstention as just another action when our confidence-rated predictors are indeed specialists. For this problem, we also present subroutines that significantly speed up CBA.
For a more detailed discussion of related work, refer to Appendix A.

2 Problem formulation and notation

We consider the classic problem of prediction with expert advice under bandit feedback. In this problem we have $K+1$ *actions*, E *experts*, and T *trials*. On each trial t:

1. Each expert suggests, to the learner, a probability distribution over the $K+1$ actions.
2. The learner selects an action a_t.
3. The reward incurred by action a_t on trial t (which is in $[-1,1]$ and is selected by Nature before the trial) is revealed to the learner.

The aim of the learner is to maximize the cumulative reward obtained by its selected actions. As discussed in Section 1, we consider the case in which there is an action (the abstention action) that incurs zero reward on every trial.

We denote our action set by $[K] \cup \{\square\}$ where $\square$ is the abstention action. For each trial $t \in [T]$ we define the vector $\boldsymbol{r}_t \in [-1,1]^K$ such that for all $a \in [K]$, $r_{t,a}$ is the reward obtained by action a on trial t. Moreover, we define $r_{t,\square} := 0$ which is the reward of the abstention action $\square$.

It will be useful for us to represent probability distributions over the actions by vectors in the set:

$$\mathcal{A} := \{\boldsymbol{s} \in [0,1]^K \mid \|\boldsymbol{s}\|_1 \leq 1\}\,.$$

Any vector $\boldsymbol{s} \in \mathcal{A}$ represents the probability distribution over actions which assigns, for all $a \in [K]$, a probability of s_a to action a, and assigns a probability of $1 - \|\boldsymbol{s}\|_1$ to the abstention action $\square$, where $\|\boldsymbol{s}\|_1$ denotes 1-norm of $\boldsymbol{s}$. We write $a \sim \boldsymbol{s}$ to represent that an action a is drawn from this probability distribution. We will refer to the elements of the set $\mathcal{A}$ as *stochastic actions*.

A *policy* is any element of $\mathcal{A}^T$ (noting that any such policy is a matrix in $[0,1]^{T\times K}$). Any policy $\boldsymbol{e} \in \mathcal{A}^T$ defines a stochastic sequence of actions: on every trial $t \in [T]$ an action $a \in [K] \cup \{\square\}$ being drawn as $a \sim \boldsymbol{e}_t$. Note that if the learner plays according to a policy $\boldsymbol{e} \in \mathcal{A}^T$, then on each trial t it obtains an expected reward of $\boldsymbol{r}_t \cdot \boldsymbol{e}_t$, where the operator $\cdot$ denotes the dot product. Note that each expert is equivalent to a policy. Thus, for all $i \in [E]$ we denote the i-th expert by $\boldsymbol{e}^i \in \mathcal{A}^T$. At the start of each trial $t \in [T]$, the learner views the sequence $\langle \boldsymbol{e}^i_t \mid i \in [E] \rangle$.

We can also view the experts as *confidence-rated predictors* over the set $[K]$: for each $i \in [E]$ and $t \in [T]$, the vector $\boldsymbol{e}^i_t$ can be viewed as suggesting the probability distribution $\boldsymbol{e}^i_t / \|\boldsymbol{e}^i_t\|_1$ over $[K]$, but with confidence $\|\boldsymbol{e}^i_t\|_1$. We denote this confidence by $c_{t,i} := \|\boldsymbol{e}^i_t\|_1$ and write $\boldsymbol{c}_t := (c_{t,1}, \ldots, c_{t,E})$.

In this work, we will refer to the *unnormalized relative entropy* defined by:

$$\Delta(\boldsymbol{u}, \boldsymbol{v}) := \sum_{i \in [E]} u_i \ln\left(\frac{u_i}{v_i}\right) - \|\boldsymbol{u}\|_1 + \|\boldsymbol{v}\|_1$$

for any $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^E_+$. We will also use the Iverson bracket notation $[\![\textsc{Pred}]\!]$ as the indicator function, meaning that it is equal to 1 if $\textsc{Pred}$ is true, and 0 otherwise. All the proofs are in the Appendix.

3 Main result

Our main result is represented by the bound on the cumulative reward of our algorithm CBA. We note that any *weight* vector $\boldsymbol{u} \in \mathbb{R}^E_+$ induces a matrix $\boldsymbol{\pi}(\boldsymbol{u}) \in \mathbb{R}^{T \times K}_+$ defined by

$$\boldsymbol{\pi}(\boldsymbol{u}) := \sum_{i \in [E]} u_i \boldsymbol{e}^i,$$

which is the linear combination of the experts with coefficients given by $\boldsymbol{u}$. However, only some of such linear combinations generate valid policies. Thus, we define

$$\mathcal{V} := \{\boldsymbol{u} \in \mathbb{R}^E_+ \mid \boldsymbol{\pi}(\boldsymbol{u}) \in \mathcal{A}^T\}$$

as the set of all weight vectors that generate valid policies. Particularly, note that $\boldsymbol{u} \in \mathcal{V}$ if and only if, on every trial t, the weighted sum of the confidences $\boldsymbol{u} \cdot \boldsymbol{c}_t$ is no greater than one. Given some $\boldsymbol{u} \in \mathcal{V}$, we define

$$\rho(\boldsymbol{u}) := \sum_{t \in [T]} \boldsymbol{r}_t \cdot \boldsymbol{\pi}_t(\boldsymbol{u}),$$

which would be the expected cumulative reward of the learner if it was to follow the policy $\boldsymbol{\pi}(\boldsymbol{u})$. We point out that the learner does not know $\mathcal{V}$ or the function $\boldsymbol{\pi}$ a-priori.

The following theorem (proved in Appendix B) allows us to bound the regret of CBA with respect to any valid linear combination $\boldsymbol{u}$ of experts.

Theorem 3.1. CBA *takes parameters $\eta \in (0, 1)$ and $\boldsymbol{w}_1 \in \mathbb{R}^E_+$. For any $\boldsymbol{u} \in \mathcal{V}$ the expected cumulative reward of* CBA *is bounded below by:*

$$\sum_{t \in [T]} \mathbb{E}[r_{t,a_t}] \geq \mathbb{E}[\rho(\boldsymbol{u})] - \frac{\Delta(\boldsymbol{u}, \boldsymbol{w}_1)}{\eta} - \eta(12K + 2)T,$$

where the expectations are with respect to the randomization of CBA*'s strategy. The per-trial time complexity of* CBA *is in $\mathcal{O}(KE)$.*

We now compare our bound to those of previous algorithms. Firstly, $\textsc{Exp4}$ can only achieve bounds relative to a $\boldsymbol{u} \in \mathcal{V}$ with $\|\boldsymbol{u}\|_1 = 1$, in which case it essentially matches our bound but with $12K + 2$ replaced by $8K + 8$. Hence, for any $\boldsymbol{u} \in \mathcal{V}$ the $\textsc{Exp4}$ bound essentially replaces the term $\rho(\boldsymbol{u})$ in our bound by $\rho(\boldsymbol{u}) / \|\boldsymbol{u}\|_1$. Note that $\|\boldsymbol{u}\|_1$ could be as high as the number of experts which implies we can dramatically outperform $\textsc{Exp4}$[1].

When viewing our experts as confidence-rated predictors, we note that previous algorithms for this setting only give bounds on a weighted sum of the per-trial rewards where the weight on each trial is $\boldsymbol{u} \cdot \boldsymbol{c}_t$ for some $\boldsymbol{u} \in \mathcal{V}$. This is only a cumulative reward bound when $\boldsymbol{u} \cdot \boldsymbol{c}_t = 1$ for all $t \in [T]$, and finding such a $\boldsymbol{u}$ is typically impossible. When there does exist $\boldsymbol{u}$ that satisfies this constraint, the reward relative to $\boldsymbol{u}$ is essentially the same as for us [Blum and Mansour, 2007]. However, there will often be another value of $\boldsymbol{u}$ that will give us a much better bound, as we show in Section 5.2.

[1]Precisely, If for each expert there exists a trial in which the confidence is 1, then we have $0 \leq \|u\|_1 \leq E$. Otherwise can be high as $0 \leq \|u\|_1 \leq E/c^*$, where $c^* = \max_{t \in [T]} c^i_t$.

Algorithm 1 CBA$(\boldsymbol{w}_1, \eta)$

For $t = 1, 2, \ldots, T$ **do**:

1. For all $i \in [E]$ receive $\boldsymbol{e}_t^i$
2. For all $i \in [E]$ set $c_{t,i} \leftarrow \|\boldsymbol{e}_t^i\|_1$
3. **If** $\|\boldsymbol{c}_t\|_1 \leq 1$ **then**:
   (a) Set $\tilde{\boldsymbol{w}}_t \leftarrow \boldsymbol{w}_t$
4. **Else**:
   (a) By interval bisection find $\lambda > 0$ such that:
   $$\sum_{i \in [E]} c_{t,i} w_{t,i} \exp(-\lambda c_{t,i}) = 1$$
   (b) For all $i \in [E]$ set $\tilde{w}_{t,i} \leftarrow w_{t,i} \exp(-\lambda c_{t,i})$
5. Set:
   $$\boldsymbol{s}_t \leftarrow \sum_{i \in [E]} \tilde{w}_{t,i} \boldsymbol{e}_t^i$$
6. Draw $a_t \sim \boldsymbol{s}_t$
7. Receive r_{t,a_t}
8. For all $a \in [K]$ set:
   $$\hat{r}_{t,a} \leftarrow 1 - [\![a = a_t]\!](1 - r_{t,a_t})/s_{t,a_t}$$
9. For all $i \in [E]$ set $w_{(t+1),i} \leftarrow \tilde{w}_{t,i} \exp(\eta \boldsymbol{e}_t^i \cdot \hat{\boldsymbol{r}}_t)$

4 The CBA algorithm

The CBA algorithm is given in Algorithm 1. In this section, we describe its derivation via a modification of the classic *mirror descent* algorithm.

Our modification of mirror descent is based on the following mathematical objects. For all $t \in [T]$ we first define:

$$\mathcal{V}_t := \{\boldsymbol{v} \in \mathbb{R}_+^E \,|\, \|\boldsymbol{\pi}_t(\boldsymbol{v})\|_1 \leq 1\}\,,$$

which is the set of all weight vectors that give rise to linear combinations producing valid stochastic actions at trial t. Given some $t \in [T]$, we define our *objective function* $\rho_t : \mathcal{V}_t \to [-1, 1]$ as

$$\rho_t(\boldsymbol{v}) := \boldsymbol{r}_t \cdot \boldsymbol{\pi}(\boldsymbol{v}) \text{ for all } \boldsymbol{v} \in \mathcal{V}_t.$$

Like mirror descent, CBA maintains, on each trial $t \in [T]$, a weight vector $\boldsymbol{w}_t \in \mathbb{R}_+^E$. However, unlike mirror descent on the simplex, we do not keep $\boldsymbol{w}_t$ normalized, but we will instead project it into $\mathcal{V}_t$ at the start of trial t, producing a vector $\tilde{\boldsymbol{w}}_t$. Also, unlike mirror descent, CBA does not use the actual gradient (which it does not know) of ρ_t at $\tilde{\boldsymbol{w}}_t$, but (inspired by the EXP3 algorithm) uses an unbiased estimator instead. Specifically, on each trial $t \in [T]$, CBA does the following:

1. Set $\tilde{\boldsymbol{w}}_t \leftarrow \operatorname{argmin}_{\boldsymbol{v} \in \mathcal{V}_t} \Delta(\boldsymbol{v}, \boldsymbol{w}_t)$.
2. Randomly construct a vector $\boldsymbol{g}_t \in \mathbb{R}^E$ such that $\mathbb{E}[\boldsymbol{g}_t] = \nabla \rho_t(\tilde{\boldsymbol{w}}_t)$.
3. Set $\boldsymbol{w}_{t+1} \leftarrow \operatorname{argmin}_{\boldsymbol{v} \in \mathbb{R}_+^E}(\eta \boldsymbol{g}_t \cdot (\tilde{\boldsymbol{w}}_t - \boldsymbol{v}) + \Delta(\boldsymbol{v}, \tilde{\boldsymbol{w}}_t))$.

This naturally raises two questions: how is a_t selected and how is $\boldsymbol{g}_t$ constructed? On each trial $t \in [T]$ we define

$$\boldsymbol{s}_t := \sum_{i \in [E]} \tilde{w}_{t,i} \boldsymbol{e}_t^i\,,$$

which is the stochastic action generated by the linear combination $\tilde{\boldsymbol{w}}_t$, and select $a_t \sim \boldsymbol{s}_t$. Note that:

$$\mathbb{E}[r_{t,a_t}] = \rho_t(\tilde{\boldsymbol{w}}_t)\,, \tag{1}$$

which confirms that ρ_t is our objective function at trial t. Once r_{t,a_t} is revealed to us we can proceed to construct the gradient estimator $\boldsymbol{g}_t$. It is important that we construct this estimator in a specific way. Inspired by EXP4 we first define a reward estimator $\hat{\boldsymbol{r}}_t$ such that for all $a \in [K]$ we have:

$$\hat{r}_{t,a} := 1 - [\![a = a_t]\!](1 - r_{t,a_t})/s_{t,a_t}\,.$$

This reward estimate is unbiased as:

$$\mathbb{E}[\hat{r}_{t,a}] = 1 - \Pr[a = a_t](1 - r_{t,a})/s_{t,a} = r_{t,a}\,.$$

We then define, for all $i \in [E]$, the component:

$$g_{t,i} := \boldsymbol{e}_t^i \cdot \hat{\boldsymbol{r}}_t\,.$$

Note that for all $i \in [E]$ we have:

$$\mathbb{E}[g_{t,i}] = \boldsymbol{e}_t^i \cdot \mathbb{E}[\hat{\boldsymbol{r}}_t] = \boldsymbol{e}_t^i \cdot \boldsymbol{r}_t = \partial_i \rho_t(\tilde{\boldsymbol{w}}_t)$$

so that $\mathbb{E}[\boldsymbol{g}_t] = \nabla \rho_t(\tilde{\boldsymbol{w}}_t)$ as required.

Now that we defined the process by which CBA operates we must show how to compute $\tilde{\boldsymbol{w}}_t$ and $\boldsymbol{w}_{t+1}$. First we show how to compute $\tilde{\boldsymbol{w}}_t$ from $\boldsymbol{w}_t$. If $\|\boldsymbol{c}_t\|_1 \leq 1$ it holds that $\boldsymbol{w}_t \in \mathcal{V}_t$ so we immediately have $\tilde{\boldsymbol{w}}_t = \boldsymbol{w}_t$. Otherwise we must find $\tilde{\boldsymbol{w}}_t \in \mathbb{R}_+^E$ that minimizes $\Delta(\tilde{\boldsymbol{w}}_t, \boldsymbol{w}_t)$ subject to the constraint: $\sum_{i\in[E]} \tilde{w}_{t,i} c_{t,i} = 1$, which is equivalent to the constraint that $\|\boldsymbol{\pi}(\tilde{\boldsymbol{w}}_t)\|_1 = 1$. Hence, by Lagrange's theorem there exists λ such that:

$$\nabla_{\tilde{\boldsymbol{w}}_t}\left(\Delta(\tilde{\boldsymbol{w}}_t, \boldsymbol{w}_t) + \lambda \sum_{i\in[E]} \tilde{w}_{t,i} c_{t,i}\right) = 0$$

which is solved by setting, for all $i \in [E]$:

$$\tilde{w}_{t,i} := w_{t,i} \exp(-\lambda c_{t,i})\,.$$

The constraint is then satisfied if λ is such that:

$$\sum_{i\in[E]} c_{t,i} w_{t,i} \exp(-\lambda c_{t,i}) = 1\,.$$

Since this function is monotonic decreasing, λ can be found by interval bisection.

Turning to the computation of $\boldsymbol{w}_{t+1}$, since it is unconstrained it is found by the equation:

$$\nabla_{\boldsymbol{w}_{t+1}}(\boldsymbol{g}_t \cdot \boldsymbol{w}_{t+1} + \eta^{-1}\Delta(\boldsymbol{w}_{t+1}, \tilde{\boldsymbol{w}}_t)) = 0\,.$$

which is solved by setting, for all $i \in [E]$:

$$w_{(t+1),i} := \tilde{w}_{t,i} \exp(\eta g_{t,i})\,. \tag{2}$$

5 Adversarial contextual bandits with abstention

One main application of CBA is in the problem of adversarial contextual bandits with a finite context set. In this problem, we have a finite set of *contexts* $\mathcal{X}$. A-priori nature selects a sequence:

$$\langle (x_t, \boldsymbol{r}_t) \in \mathcal{X} \times [-1,1]^K \,|\, t \in [T] \rangle\,,$$

but does not reveal it to the learner. For all $t \in [T]$ we define $r_{t,\square} := 0$. On each trial $t \in [T]$ the learner observes the context x_t, selects an action $a_t \in [K] \cup \{\square\}$, and sees and incurs the reward $r_{t,a_t} \in [-1,1]$.

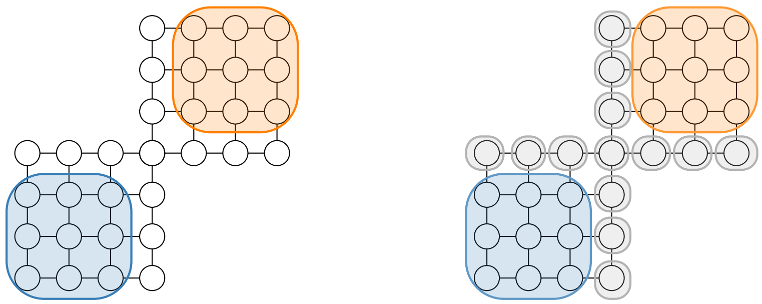
We will assume that we are given, a-priori, a set $\mathcal{B} \subseteq 2^{\mathcal{X}}$ that we call the *basis*. We call each element of $\mathcal{B}$ a *basis element* (which is a set of contexts). We will later introduce various potential bases, determined by the nature of the context's structure: points within a metric space, nodes within a graph, and beyond. Importantly, our method is capable of accommodating any type of basis and, thus, any potential inductive bias that might be present in the data.

Given our basis we run our algorithm CBA with each expert corresponding to a pair $(B,k) \in \mathcal{B}\times[K]$. The expert corresponding to each pair (B,k) will deterministically choose action k when the current context x_t is in B, and abstain otherwise. We can therefore state the following theorem (Proved in Appendix C)

Corollary 5.1. *Given any basis $\mathcal{B}$ of cardinality N and any $M \in \mathbb{N}$ we can implement* CBA *such that for any sequence of disjoint basis elements $\langle B_j \,|\, j \in [M]\rangle$ with corresponding actions $\langle b_j \in [K] \,|\, j \in [M]\rangle$ we have:*

$$\sum_{t\in[T]} \mathbb{E}[r_{t,a_t}] \geq \sum_{t\in[T]} \sum_{j\in[M]} [\![x_t \in B_j]\!] r_{t,b_j} - \sqrt{2M\ln(N)(6K+1)T}\,.$$

The per-trial time complexity of this implementation of CBA *is in $\mathcal{O}(KN)$.*



(a) Two foreground classes and background as abstained.

(b) Two foreground classes and the background as another one.

Figure 1: Illustrative example of abstention where we cover the foreground and background classes with metric balls. We consider two clusters (blue and orange) as the foreground and one background class (white), using the shortest path d_∞ metric. Using abstention, we can cover two clusters with one ball for each and abstain the background with no balls required (Fig. 1(a)). In contrast, if we treat the background class as another class, it would require significantly more balls to cover the background class, as seen other 10 gray balls in Fig. 1(b). This increase in the number of balls would lead to a significantly worse bound that involves the number of balls.

We briefly comment on the term:

$$\sum_{j\in[M]} [\![x_t \in B_j]\!] r_{t,b_j} \,,$$

that appears in the theorem statement. If x_t does not belong to any of the sets in $\langle B_j \mid j \in [M]\rangle$ then this term is equal to zero (which is the reward of abstaining). Otherwise, since the sets are disjoint, x_t belongs to exactly one of them and the term is equal to the reward induced by the action that corresponds to that set. In other words, the total cumulative reward is bounded relative to that of the policy that abstains whenever x_t is outside the union of the sets and otherwise selects the action corresponding to the set that x_t lies in.

Note the vast improvement of our reward bound over that of SPECIALISTEXP with abstention as one of the actions. Let's assume our context set is a metric space and our basis is the set of all balls. In order to get a reward bound for SPECIALISTEXP, the sets in which the specialists are awake must partition the set $\mathcal{X}$. This means that we must add to our M balls a disjoint covering (by balls) the complement of the union of the original M balls. Note that the added balls correspond to the sets in which the specialists predicting the abstention action are awake. Typically this would require a huge number of balls so that the total number of specialists is huge (much larger than M); this huge number of specialists essentially replaces the term M in our reward bound (we illustrate an example in Figure 1).

Furthermore, in Appendix F, we show that the same implementation of CBA is capable of learning a weighted set of *overlapping* basis elements, as long as the sum of the weights of the basis elements covering any context is bounded above by one, which SPECIALISTEXP cannot do in general.

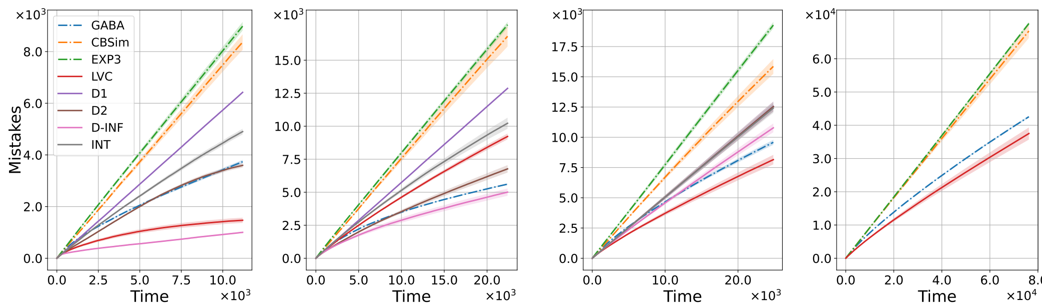
As we will see below, the practical bases we propose have a moderate size of $|\mathcal{B}| = \mathcal{O}(|\mathcal{X}|^2)$ leading to a per-step runtime of $\mathcal{O}(K|\mathcal{X}|^2)$ for CBA in this contextual bandit problem. In Section 5.2, we show how to significantly improve the runtime for a broad family of bases.

5.1 A lower bound

In this section, we show that CBA is, up to an $\mathcal{O}(\ln(|\mathcal{B}|))$ factor, essentially best possible on this contextual bandit problem:

Proposition 5.2. *Take any learning algorithm. Given any basis $\mathcal{B}$ and any $M \in \mathbb{N}$, for any sequence of disjoint basis elements $\langle B_j \mid j \in [M]\rangle$ there exists a sequence of corresponding actions $\langle b_j \in [K] \mid j \in [M]\rangle$ such that an adversary can force:*

$$\sum_{t\in[T]}\sum_{j\in[M]} [\![x_t \in \mathcal{B}_j]\!] r_{t,b_j} - \sum_{t\in[T]} \mathbb{E}[r_{t,a_t}] \in \Omega\left(\sqrt{MKT}\right) .$$



(a) Stochastic Block Model (b) Gaussian graph (c) Cora graph (d) LastFM Asia graph

Figure 2: Results regarding the number of mistakes over time. In this context, D1, D2, and D-INF represent the p-norm bases, LVC represents the community detection basis, and INT represents the interval basis. The baselines, EXP3 for each context, Contextual Bandit with similarity, and GABA-II, are denoted as EXP3, CBSim, and GABA, respectively, and are represented with dashed lines.

5.2 Efficient learning with balls

In practice we can often quantify the similarity between any pair of contexts. That is, the contexts form a metric space, equipped with a *distance* function $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ known to the learner a-priori. For example, contexts could have feature vectors in $\mathbb{R}^m$ (and the metric is the standard Euclidean distance or cosine similarity) or be nodes in a graph with the metric given by the shortest-path distance. A natural basis for this situation is the set of metric *balls*. Specifically, a ball is any set $B \subseteq \mathcal{X}$ in which there exists some $x \in \mathcal{X}$ and $\delta \in \mathbb{R}_+$ with:

$$B = \{z \in \mathcal{X} \,|\, d(x, z) \leq \delta\}\,.$$

For this broad family of bases[2] we can achieve the following speed-up, relying on a sophisticated data structure based on binary trees.

Theorem 5.3. *Let $N := |\mathcal{X}|$. Given any $M \in \mathbb{N}$ we can implement* CBA *such that for any sequence of disjoint balls $\langle B_j \,|\, j \in [M]\rangle$ with corresponding actions $\langle b_j \in [K] \,|\, j \in [M]\rangle$ we have:*

$$\sum_{t\in[T]} \mathbb{E}[r_{t,a_t}] \geq \sum_{t\in[T]} \sum_{j\in[M]} [\![x_t \in B_j]\!] r_{t,b_j} - \sqrt{4M\ln(N)(6K+1)T}\,.$$

The per-trial time complexity of this implementation of CBA *is in $\mathcal{O}(KN\ln(N))$.*

As there are at most $\mathcal{O}(N^2)$ metric balls, this improves the runtime of the direct CBA implementation from $\mathcal{O}(KN^2)$ to $\mathcal{O}(KN\ln(N))$, that is almost linear per step. All the details are in Appendix D.

6 Preliminary experiments

As mentioned above, the bases used in our algorithm can be constructed arbitrarily, allowing to encompass different inductive biases based on applications. Thus, we consider some representative bases used on learning tasks on graphs before, each leading to different inductive priors on the contexts. We provide a short description of the bases here and refer to Appendix G for more details.

Effective p-resistance basis d_p: Balls given by the metric

$$d_p(i,j) := \left(\min_{\substack{\boldsymbol{u}\in\mathbb{R}^N \\ u_i - u_j = 1}} \sum_{s,t\in V} |u_s - u_t|^p\right)^{-1/p}.$$

We use d_1, d_2, and d_∞ [Herbster and Lever, 2009].

[2] Actually, we require a weaker condition. We only use the fact that for each context $z \in \mathcal{X}$ we have a set $\mathcal{B}_z = \{B_1^z, \ldots, B_\ell^z\}$ of monotonically increasing basis elements, that is, $B_i^z \subseteq B_j^z$ for $i < j$, and the whole basis is formed by the union of these $\mathcal{B} = \bigcup_{z\in\mathcal{X}} \mathcal{B}_z$.

Louvain method basis (LVC): Communities returned by the Louvain method [Blondel et al., 2008], processed by the greedy peeling algorithm [Lanciano et al., 2024].

Geodesic intervals basis (INT): All sets of the form $I(x, y) := \{z \in \mathcal{X} \mid z \text{ is on a shortest } x\text{-}y \text{ path}\}$ for all $x, y \in \mathcal{X}$ [Pelayo, 2013, Thiessen and Gärtner, 2021].

Let N be the cardinality of $|\mathcal{X}|$. For all three basis types, we immediately get an $\mathcal{O}(KN^2)$ runtime per step of CBA as there are $\mathcal{O}(N^2)$ basis elements. Moreover, for d_p balls and the LVC basis we can use the more efficient $\mathcal{O}(KN \ln N)$ implementation through Theorem 5.3. We empirically evaluate our approach in the context of online multi-class node classification on a given graph with bandit feedback. At each time step, the algorithm is presented with a node chosen uniformly at random and must either predict an action from the set of possible actions $[K]$ or abstain. The node can accept (resulting in a positive reward) or reject (resulting in a negative reward) the suggestion based on its preferred class with a certain probability.

We compare our approach CBA using each of these bases on real-world and artificial graphs against the following baselines: an implementation of CONTEXTUALBANDIT from Slivkins [2011], the GABA-II algorithm proposed by Herbster et al. [2021], and an EXP3 instance for each data point. We use the following graphs for evaluation.

Stochastic block model. This graph, inspired by Holland et al. [1983], is generated by spawning an arbitrary number of disjoint cliques representing the foreground classes. Then an arbitrary number of background points are generated and connected to every possible point with a low probability. In Figure 2(a) are displayed the results for the case of $F = 160$ nodes for each foreground class and $B = 480$ nodes for the background class. Connecting each node of the background class with a probability of $1/\sqrt{FB}$.

Gaussian graph. The points on this graph are generated in a two-dimensional space using five different Gaussian distributions with zero mean. Four of them are positioned at the corners of the unit square, representing the foreground classes and having a relatively low standard deviation. Meanwhile, the fifth distribution, representing the background class, is centered within the square and is characterized by a larger standard deviation. The points are linked in a k-nearest neighbors graph. In Figure 2(b) are displayed the results for 160 nodes for each foreground class and a standard deviation of 0.2, 480 nodes for the background class with a standard deviation of 1.75, along with a 7-nearest neighbors graph.

Real-world dataset. We tested our approach on the Cora dataset [Sen et al., 2008] and the LastFM Asia dataset [Leskovec and Krevl, 2014]. While both of these graphs contain both features and a graph, we exclusively utilized the largest connected component of each graph, resulting in 2485 nodes and 5069 edges for the Cora graph and 7624 nodes and 27806 edges for the LastFM Asia graph. Subsequently, we randomly chose a subset of three out of the original seven and eighteen classes, respectively, to serve as the background class. Additionally, we selected 15% of the nodes from the foreground classes randomly to represent noise points, and we averaged the results over multiple runs, varying the labels chosen for noise. Both in Figures 2(c) and 2(d) we averaged over 5 different label sets as noise. For the LastFM Asia graph, we exclusively tested the LVC bases, as it is the most efficient one to compute given the large size of the graph.

Results. The results from both synthetically generated tests (Figures 2(a) and 2(b)) demonstrate the superiority of our method when compared to the baselines. In particular, d_∞-balls delivered exceptional results for both graphs, implying that d_∞-balls effectively cover the foreground classes as expected. For the Cora dataset (Figure 2(c)), we observed that our method outperforms GABA-II only when employing the community detection basis. This similarity in performance is likely attributed to the dataset's inherent lack of noise. Worth noting that the method we employed to inject noise into the dataset may not have been the optimal choice for this specific context. For the LastFM Asia dataset, our objective was to assess the practical feasibility of the model on a larger graph. We tested the LVC bases as they were the most promising and most efficient to compute. We outperform the baselines in our evaluation as shown in Figure 2(d) and further discussed in Appendix H.

In summary, our first results confirm what we expected: our approach excels when we choose basis functions that closely match the context's structure. However, it also encounters difficulties when the chosen basis functions are not a good fit for the context. In Appendix H, the results for a wide range of different parameters used to generate the previously described graphs are displayed.

References

Morteza Alamgir and Ulrike von Luxburg. Phase transition in the family of p-resistances. In *Proc. NIPS*, pages 379–387, 2011.

Peter Auer and Philip M Long. Structural results about on-line learning models with and without queries. *Mach. Learn.*, 36:147–181, 1999.

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

Amir Beck. *First-Order Methods in Optimization*. SIAM, 2017.

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.*, 2008(10):P10008, 2008.

Avrim Blum and Yishay Mansour. From external to internal regret. *J. Mach. Learn. Res.*, 8(6), 2007.

Marco Bressan, Nicolò Cesa-Bianchi, Silvio Lattanzi, and Andrea Paudice. Exact recovery of clusters in finite metric spaces using oracle queries. In *Proc. COLT*, 2021.

C Chow. On optimum recognition error and reject tradeoff. *IEEE Trans. Inf. Theory*, 16(1):41–46, 1970.

Chi-Keung Chow. An optimum character recognition system using decision functions. *IRE Trans. Electron. Comput.*, EC-6(4):247–254, 1957.

Corinna Cortes, Giulia DeSalvo, Claudio Gentile, Mehryar Mohri, and Scott Yang. Online learning with abstention. In *Proc. ICML*, pages 1059–1067, 2018.

Corinna Cortes, Giulia DeSalvo, Claudio Gentile, Mehryar Mohri, and Ningshan Zhang. Online learning with dependent stochastic feedback graphs. In *Proc. ICML*, pages 2154–2163, 2020.

Peter G Doyle and J Laurie Snell. *Random Walks and Electric Networks*, volume 22 of *The Carus Mathematical Monographs*. American Mathematical Society, 1984.

Robert W Floyd. Algorithm 97: shortest path. *Commun. ACM*, 5(6):345, 1962.

Santo Fortunato. Community detection in graphs. *Phys. Rep.*, 486(3):75–174, 2010.

Yoav Freund, Robert E Schapire, Yoram Singer, and Manfred K Warmuth. Using and combining predictors that specialize. In *Proc. STOC*, pages 334–343, 1997.

Pierre Gaillard, Gilles Stoltz, and Tim Van Erven. A second-order bound with excess losses. In *Proc. COLT*, pages 176–196, 2014.

Ralph E Gomory and Tien Chung Hu. Multi-terminal network flows. *J. Soc. Ind. Appl. Math.*, 9(4): 551–570, 1961.

Kilian Hendrickx, Lorenzo Perini, Dries Van der Plas, Wannes Meert, and Jesse Davis. Machine learning with a reject option: A survey. *arXiv preprint arXiv:2107.11277*, 2021.

Mark Herbster and Guy Lever. Predicting the labelling of a graph via minimum p-seminorm interpolation. In *Proc. COLT*, 2009.

Mark Herbster, Stephen Pasteris, Fabio Vitale, and Massimiliano Pontil. A gang of adversarial bandits. In *Proc. NeurIPS*, pages 2265–2279, 2021.

Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Soc. Netw.*, 5(2):109–137, 1983.

Tommaso Lanciano, Atsushi Miyauchi, Adriano Fazzone, and Francesco Bonchi. A survey on the densest subgraph problem and its variants. *ACM Comput. Surv.*, 56(8):1–40, 2024.

Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.

Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data`, June 2014.

Haipeng Luo and Robert E Schapire. Achieving all with no parameters: Adanormalhedge. In *Proc. COLT*, pages 1286–1304, 2015.

Gergely Neu and Nikita Zhivotovskiy. Fast rates for online prediction with abstention. In *Proc. COLT*, pages 3030–3048, 2020.

Mark E.J. Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69(2):026113, 2004.

Ignacio M. Pelayo. *Geodesic Convexity in Graphs*. Springer, 2013.

Shota Saito and Mark Herbster. Multi-class graph clustering via approximated effective p-resistance. In *Proc. ICML*, pages 29697–29733, 2023.

Nicolas Schreuder and Evgenii Chzhen. Classification with abstention but without disparities. In *Proc. UAI*, pages 1227–1236, 2021.

Yevgeny Seldin and Gábor Lugosi. A lower bound for multi-armed bandits with expert advice. In *Proc. EWRL*, volume 2, page 7, 2016.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI Mag.*, 29(3):93–93, 2008.

Aleksandrs Slivkins. Contextual bandits with similarity information. In *Proc. COLT*, pages 679–702, 2011.

Maximilian Thiessen and Thomas Gärtner. Active learning of convex halfspaces on graphs. In *Proc. NeurIPS*, 2021.

Vincent A Traag. Faster unfolding of communities: Speeding up the louvain algorithm. *Phys. Rev. E*, 92(3):032801, 2015.

Marcel LJ van De Vel. *Theory of Convex Structures*. Elsevier, 1993.

A Additional related works

The non-stochastic multi-armed bandit problem, initially introduced by Auer et al. [2002], has been a subject of significant research interest. Auer et al. [2002] also introduced the multi-armed bandit problem with expert advice, introducing the EXP4 algorithm. EXP4 evolved the field of multi-armed bandits to encompass more complex scenarios, particularly the contextual bandit [Lattimore and Szepesvári, 2020]. Contextual bandits are an extension of the classical multi-armed bandit framework, where an agent makes a sequence of decisions while taking into account contextual information. We also mention that our work is also related to the multi-class classification with bandit feedback, called *weak reinforcement* [Auer and Long, 1999]. An action in our bandit setting corresponds to a class in the multi-class classification framework.

As discussed in the introduction, a key aspect of this work is the option to abstain from making any decision. In the batch setting [Chow, 1957, 1970], this option is usually referred to as "rejection". These works study whether to use or reject a specific model prediction based on specific requests (see Hendrickx et al. [2021] for a survey). In online learning, "rejection" can be the possibility of abstention by the learner. These works usually rely on a cost associated with the abstention action. Neu and Zhivotovskiy [2020] studied the magnitude of the cost associated with abstention in an expert setting with bounded losses. They state that if the cost is lower than half of the amplitude of the interval of the loss, it is possible to derive bounds that are independent of the time. In Cortes et al. [2018], a non-contextual and partial information setting with the option of abstention is studied. The sequel model [Cortes et al., 2020] regards this model as a special case of their stochastic feedback graph model. Schreuder and Chzhen [2021] studied the fairness setting when using the option of abstaining as it may lead to discriminatory predictions.

B CBA analysis

Here we prove Theorem 3.1 from the modification of mirror descent (and the specific construction of $\boldsymbol{g}_t$) given in Section 4. Whenever we take expectations in this analysis they are over the draw of a_t from $\boldsymbol{s}_t$ for some $t \in [T]$. As for mirror descent, our analysis hinges on the following classic lemma:

Lemma B.1. *Given any convex set $\mathcal{C} \subseteq \mathbb{R}_+^E$, any convex function $\xi : \mathbb{R}_+^E \to \mathbb{R}$, any $\boldsymbol{q} \in \mathcal{C}$ and any $\boldsymbol{z} \in \mathbb{R}_+^E$ with:*

$$\boldsymbol{q} = \operatorname{argmin}_{\boldsymbol{v} \in \mathcal{C}} (\xi(\boldsymbol{v}) + \Delta(\boldsymbol{v}, \boldsymbol{z}))\,,$$

then for all $\boldsymbol{u} \in \mathcal{C}$ we have:

$$\xi(\boldsymbol{u}) + \Delta(\boldsymbol{u}, \boldsymbol{z}) \geq \xi(\boldsymbol{q}) + \Delta(\boldsymbol{u}, \boldsymbol{q})\,.$$

Proof. Theorem 9.12 in Beck [2017] shows that the theorem holds if Δ is Bregman divergence. In our case Δ is indeed a Bregman divergence: that of the convex function $f : \mathbb{R}_+^E \to \mathbb{R}$ for all $\boldsymbol{v} \in \mathbb{R}_+^E$ defined by:

$$f(\boldsymbol{v}) := \sum_{i \in [E]} v_i \ln(v_i),$$

which concludes the proof. □

Choose any $\boldsymbol{u} \in \mathcal{V}$ and $t \in [T]$. We immediately have $\mathcal{V} \subseteq \mathcal{V}_t$ by definition, and therefore $\boldsymbol{u} \in \mathcal{V}_t$. Hence, by setting ξ such that $\xi(\boldsymbol{v}) := 0$ for all $\boldsymbol{v} \in \mathbb{R}_+^E$, setting $\mathcal{C} \in \mathcal{V}_t$ and setting $\boldsymbol{z} = \boldsymbol{w}_t$ in Lemma B.1 we have $\boldsymbol{q} = \tilde{\boldsymbol{w}}_t$ so that:

$$\Delta(\boldsymbol{u}, \boldsymbol{w}_t) \geq \Delta(\boldsymbol{u}, \tilde{\boldsymbol{w}}_t)\,. \tag{3}$$

Alternatively, by setting ξ such that $\xi(\boldsymbol{v}) := \eta \boldsymbol{g}_t \cdot (\tilde{\boldsymbol{w}}_t - \boldsymbol{v})$ for all $\boldsymbol{v} \in \mathbb{R}_+^E$, setting $\mathcal{C} = \mathbb{R}_+^E$ and setting $\boldsymbol{z} = \tilde{\boldsymbol{w}}_t$ in Lemma B.1 we have $\boldsymbol{q} = \boldsymbol{w}_{t+1}$ so that:

$$\eta \boldsymbol{g}_t \cdot (\tilde{\boldsymbol{w}}_t - \boldsymbol{u}) + \Delta(\boldsymbol{u}, \tilde{\boldsymbol{w}}_t) \geq \eta \boldsymbol{g}_t \cdot (\tilde{\boldsymbol{w}}_t - \boldsymbol{w}_{t+1}) + \Delta(\boldsymbol{u}, \boldsymbol{w}_{t+1})\,. \tag{4}$$

Since $\mathbb{E}[\boldsymbol{g}_t] = \nabla \rho_t(\tilde{\boldsymbol{w}}_t)$ and ρ_t is linear we have:

$$\mathbb{E}[\boldsymbol{g}_t \cdot (\tilde{\boldsymbol{w}}_t - \boldsymbol{u})] = \rho_t(\tilde{\boldsymbol{w}}_t) - \rho_t(\boldsymbol{u})\,. \tag{5}$$

In what follows we use the fact that for all $x \leq 1$ we have:

$$x(1 - \exp(x)) \geq -2x^2\,. \tag{6}$$

For all $i \in [E]$, we have, by definition, that $g_{t,i} = \boldsymbol{e}_t^i \cdot \hat{\boldsymbol{r}}_t$ so by Equation (2) we have:

$$\boldsymbol{g}_t \cdot (\tilde{\boldsymbol{w}}_t - \boldsymbol{w}_{t+1}) = \sum_{i \in [E]} \tilde{w}_{t,i} \boldsymbol{e}_t^i \cdot \hat{\boldsymbol{r}}_t (1 - \exp(\eta \boldsymbol{e}_t^i \cdot \hat{\boldsymbol{r}}_t)) \,.$$

Since, for all $a \in [K]$, we have $\hat{r}_{t,a} \leq 1$ and hence, as $\eta < 1$ and, for all $i \in [E]$ we have $\|\boldsymbol{e}_t^i\|_1 \leq 1$, we can invoke Equation (6), which gives us:

$$\eta \boldsymbol{g}_t \cdot (\tilde{\boldsymbol{w}}_t - \boldsymbol{w}_{t+1}) \geq -2 \sum_{i \in [E]} \tilde{w}_{t,i} (\eta \boldsymbol{e}_t^i \cdot \hat{\boldsymbol{r}}_t)^2 \,. \tag{7}$$

By definition of $\hat{\boldsymbol{r}}_t$ we have, for all $i \in [E]$, that:

$$\boldsymbol{e}_t^i \cdot \hat{\boldsymbol{r}}_t = \|\boldsymbol{e}_t^i\|_1 + e_{t,a_t}^i (1 - r_{t,a_t}) / s_{t,a_t} \leq c_{t,i} + 2 e_{t,a_t}^i / s_{t,a_t}$$

so that since, for all $a \in [K]$, we have $\Pr[a_t = a] = s_{t,a}$ we also have:

$$\mathbb{E}[(\boldsymbol{e}_t^i \cdot \hat{\boldsymbol{r}}_t)^2] \leq c_{t,i}^2 + \sum_{a \in [K]} (2 e_{t,a}^i c_{t,i} + 4 (e_{t,a}^i)^2 / s_{t,a}) \,. \tag{8}$$

Since, for all $i \in [E]$ and $a \in [K]$, we have $e_{t,a}^i \leq 1$ and $c_{t,i} \leq 1$ and hence also $c_{t,i}^2 \leq c_{t,i}$ we then have:

$$\mathbb{E}[(\boldsymbol{e}_t^i \cdot \hat{\boldsymbol{r}}_t)^2] \leq (2K + 1) c_{t,i} + 4 \sum_{a \in [K]} e_{t,a}^i / s_{t,a} \,. \tag{9}$$

Note that since $\tilde{\boldsymbol{w}}_t \in \mathcal{V}_t$ we have:

$$\sum_{i \in [E]} \tilde{w}_{t,i} c_{t,i} \leq 1 \,. \tag{10}$$

Also, by definition of $\boldsymbol{s}_t$ we have:

$$\sum_{i \in [E]} \tilde{w}_{t,i} \sum_{a \in [K]} e_{t,a}^i / s_{t,a} = \sum_{a \in [K]} \frac{1}{s_{t,a}} \sum_{i \in [E]} \tilde{w}_{t,i} e_{t,a}^i = \sum_{a \in [K]} \frac{1}{s_{t,a}} s_{t,a} = K \,. \tag{11}$$

Multiplying Inequality (9) by $\tilde{w}_{t,i}$, summing over all $i \in [E]$, and then substituting in Inequality (10) and Equation (11) gives us:

$$\sum_{i \in [E]} \tilde{w}_{t,i} \mathbb{E}[(\boldsymbol{e}_t^i \cdot \hat{\boldsymbol{r}}_t)^2] \leq (2K + 1) + 4K = 6K + 1 \,. \tag{12}$$

Taking expectations on Inequality (7) and substituting in Inequality (12) (after taking expectations) gives us:

$$\mathbb{E}[\eta \boldsymbol{g}_t \cdot (\tilde{\boldsymbol{w}}_t - \boldsymbol{w}_{t+1})] \geq -\eta^2 (12K + 2) \,. \tag{13}$$

Taking expectations (over the draw $a_t \sim \boldsymbol{s}_t$) on Inequality (4), substituting in Inequalities (3), (5) and (13), and then rearranging gives us:

$$\Delta(\boldsymbol{u}, \boldsymbol{w}_t) - \mathbb{E}[\Delta(\boldsymbol{u}, \boldsymbol{w}_{t+1})] \geq \eta(\rho_t(\boldsymbol{u}) - \rho_t(\tilde{\boldsymbol{w}}_t)) - \eta^2 (12K + 2) \,.$$

Summing this inequality over all $t \in [T]$, taking expectations (over the entire sequence of action draws) and noting that $\Delta(\boldsymbol{u}, \boldsymbol{w}_{T+1}) > 0$ gives us:

$$\Delta(\boldsymbol{u}, \boldsymbol{w}_1) \geq \eta \sum_{t \in [T]} \mathbb{E}[\rho_t(\boldsymbol{u}) - \rho_t(\tilde{\boldsymbol{w}}_t)] - \eta^2 (12K + 2) T \,.$$

Substituting in Equation (1) and rearranging then gives us, by definition of ρ and ρ_t, the required goal:

$$\sum_{t \in [T]} \mathbb{E}[r_{t,a_t}] \geq \mathbb{E}[\rho(\boldsymbol{u})] - \Delta(\boldsymbol{u}, \boldsymbol{w}_1)/\eta - \eta(12K + 2) T \,.$$

■

C Corollary proof

Corollary C.1. *Given any basis $\mathcal{B}$ of cardinality N and any $M \in \mathbb{N}$ we can implement* CBA *such that for any sequence of disjoint basis elements $\langle B_j \,|\, j \in [M] \rangle$ with corresponding actions $\langle b_j \in [K] \,|\, j \in [M] \rangle$ we have:*

$$\sum_{t \in [T]} \mathbb{E}[r_{t,a_t}] \geq \sum_{t \in [T]} \sum_{j \in [M]} [\![x_t \in B_j]\!] r_{t,b_j} - \sqrt{2M \ln(N)(6K+1)T} \,.$$

The per-trial time complexity of this implementation of CBA *is in $\mathcal{O}(KN)$.*

Proof. The choice of experts for CBA that leads to Theorem 5.1 is defined by the set of pairs so that $E = N^2 K$ and for each $B \in \mathcal{B}$ and action $a \in [K]$ there exists an unique $i \in [E]$ such that for all $t \in [T]$ and $b \in [K]$ we have:

$$e^i_{t,b} := [\![x_t \in B]\!] [\![b = a]\!] \,.$$

By choosing $w_{1,i} = M/N^2$ for all $i \in [E]$, and choosing

$$\eta := (M \ln(N)/(6K+1)T)^{-1/2} \,,$$

Theorem 3.1 implies the reward bound in Corollary 5.1. The per-trial time complexity of a direct implementation of CBA for this set of experts would be $\mathcal{O}(KN)$. □

D Efficient implementation proof

We here prove the time complexity of Theorem 5.3. The per-trial time complexity of a direct implementation of CBA for this set of experts would be $\mathcal{O}(KN^2)$. We now show how to implement CBA in a per-trial time of only $\mathcal{O}(KN \ln(N))$. To do this first note that we can assume, without loss of generality, that for all $q, x, z \in \mathcal{X}$ with $x \neq z$ we have $d(q,x) \neq d(q,z)$ since ties can be broken arbitrarily and balls can be duplicated.

Given $x, z \in \mathcal{X}$, $a \in [K]$ and $t \in [T]$ we let $y_{t,a}(x,z) := w_{t,i}$ and $\tilde{y}_{t,a}(x,z) := \tilde{w}_{t,i}$ where i is the index of the expert corresponding to the ball-action pair with ball: $\{q \in \mathcal{X} \,|\, d(x,q) \leq d(x,z)\}$, and action a. Given $x, z \in \mathcal{X}$ let $\mathcal{E}(x,z) := \{q \in \mathcal{X} \,|\, d(x,q) \geq d(x,z)\}$. It is straightforward to derive the following equations for the quantities in CBA at trial $t \in [T]$. First we have:

$$\|\boldsymbol{c}_t\|_1 = \sum_{a \in [K]} \sum_{x \in \mathcal{X}} \sum_{z \in \mathcal{E}(x,x_t)} y_{t,a}(x,z) \,.$$

For all $x, z \in \mathcal{X}$ and $a \in [K]$ we have the following:

- If $\|\boldsymbol{c}_t\|_1 \leq 1$ or $z \notin \mathcal{E}(x,x_t)$ then $\tilde{y}_{t,a}(x,z) = y_{t,a}(x,z)$.
- If $\|\boldsymbol{c}_t\|_1 > 1$ and $z \in \mathcal{E}(x,x_t)$ then $\tilde{y}_{t,a}(x,z) = y_{t,a}(x,z)/\|\boldsymbol{c}_t\|_1$.

For all $a \in [K]$ we have:

$$s_{t,a} = \sum_{x \in \mathcal{X}} \sum_{z \in \mathcal{E}(x,x_t)} \tilde{y}_{t,a}(x,z) \,.$$

Finally, for all $x, z \in \mathcal{X}$ and $a \in [K]$ we have the following:

$$y_{(t+1),a}(x,z) = \begin{cases} \tilde{y}_{t,a}(x,z) & \text{if } z \notin \mathcal{E}(x,x_t) \,, \\ \tilde{y}_{t,a}(x,z) \exp(\eta \boldsymbol{e}^i_t \cdot \hat{\boldsymbol{r}}_t) & \text{if } z \in \mathcal{E}(x,x_t) \,. \end{cases}$$

Hence, to implement CBA we need, for each $x \in \mathcal{X}$ and $a \in [K]$, a data structure that implicitly maintains a function $h : \mathcal{X} \to \mathbb{R}^+$ and has the following two subroutines, that take parameters $q \in \mathcal{X}$ and $p \in \mathbb{R}_+$.

1. QUERY(q): Compute $\sum_{z \in \mathcal{E}(x,q)} h(z)$.
2. UPDATE(q,p): Set $h(z) \leftarrow p h(z)$ for all $z \in \mathcal{E}(x,q)$.

Algorithm 2 QUERY(q)

1. For all $i \in [n] \cup \{0\}$ let γ_i be the ancestor of q at depth i in $\mathcal{D}$
2. Set $\sigma_n \leftarrow \psi(\gamma_n)\phi(\gamma_n)$
3. Climb $\mathcal{D}$ from γ_{n-1} to γ_0. When at γ_i do as follows:
   (a) If $\gamma_{i+1} = \triangleleft(\gamma_i)$ then set $\sigma_i \leftarrow \phi(\gamma_i)(\sigma_{i+1} + \psi(\triangleright(\gamma_i))\phi(\triangleright(\gamma_i)))$
   (b) If $\gamma_{i+1} = \triangleright(\gamma_i)$ then set $\sigma_i \leftarrow \phi(\gamma_i)\sigma_{i+1}$
4. Return σ_0

Algorithm 3 UPDATE(q, p)

1. For all $i \in [n] \cup \{0\}$ let γ_i be the ancestor of q at depth i in $\mathcal{D}$
2. Descend $\mathcal{D}$ from γ_0 to γ_{n-1}. When at γ_i set:
   (a) $\phi(\triangleleft(\gamma_i)) \leftarrow \phi(\gamma_i)\phi(\triangleleft(\gamma_i))$
   (b) $\phi(\triangleright(\gamma_i)) \leftarrow \phi(\gamma_i)\phi(\triangleright(\gamma_i))$
   (c) $\phi(\gamma_i) \leftarrow 1$
3. For all $i \in [n-1] \cup \{0\}$, if $\gamma_{i+1} = \triangleleft(\gamma_i)$ then set $\phi(\triangleright(\gamma_i)) \leftarrow p\phi(\triangleright(\gamma_i))$
4. Set $\phi(\gamma_n) \leftarrow p\phi(\gamma_n)$
5. Climb $\mathcal{D}$ from γ_{n-1} to γ_0. When at γ_i set:
   $\psi(\gamma_i) \leftarrow \psi(\triangleleft(\gamma_i))\phi(\triangleleft(\gamma_i)) + \psi(\triangleright(\gamma_i))\phi(\triangleright(\gamma_i))$

Now fix $x \in \mathcal{X}$ and $a \in [K]$. Let h be as above. On each trial $t \in [T]$ and for all $z \in \mathcal{X}$, $h(z)$ will start equal to $y_{t,a}(x, z)$ and change to $\tilde{y}_{t,a}(x, z)$ and then $y_{(t+1),a}(x, z)$ by applying the UPDATE subroutine.

We now show how to implement these subroutines implicitly in a time of $\mathcal{O}(\ln(N))$ as required. Without loss of generality, assume that $N = 2^n$ for some $n \in \mathbb{N}$. Our data structure is based on a balanced binary tree $\mathcal{D}$ whose leaves are the elements of $\mathcal{X}$ in order of increasing distance from x. This implies that for any $z \in \mathcal{X}$ we have that $\mathcal{E}(x, z)$ is the set of leaves that do not lie on the left of z. Given a node $v \in \mathcal{D}$ we let $\Uparrow(v)$ be the set of ancestors of v and let $\Downarrow(v)$ be the set of all $z \in \mathcal{X}$ which are descendants of v. For any internal node v let $\triangleleft(v)$ and $\triangleright(v)$ be the left and right children of v respectively.

We maintain functions $\phi, \psi : \mathcal{D} \to \mathbb{R}_+$ such that for all $v \in \mathcal{D}$ we have:

$$\psi(v) \prod_{v' \in \Uparrow(v)} \phi(v') = \sum_{z \in \Downarrow(v)} h(z)\,. \tag{14}$$

The pseudo-code for the subroutines QUERY and UPDATE are given in Algorithms 2 and 3 respectively. We now prove their correctness. We first consider the QUERY subroutine with parameter $q \in \mathcal{X}$. From Equation (14) we see that, by (reverse) induction on $i \in [n] \cup \{0\}$, we have:

$$\sigma_i \prod_{v' \in \Uparrow(\gamma_i) \setminus \{\gamma_i\}} \phi(v') = \sum_{z \in \Downarrow(\gamma_i) \cap \mathcal{E}(x,q)} h(z)\,.$$

Since γ_0 is the root of $\mathcal{D}$, we have $\sigma_0 = \sum_{z \in \mathcal{E}(x,q)} h(z)$ as required. Now consider the UPDATE subroutine with parameters $q \in \mathcal{X}$ and $p \in \mathbb{R}_+$. Let h be the implicitly maintained function before the subroutine is called. For Equation (14) to hold after the subroutine is called we need:

$$\psi(v) \prod_{v' \in \Uparrow(v)} \phi(v') = \sum_{z \in \Downarrow(v)} h'(z)\,. \tag{15}$$

where for all $z \in \mathcal{X}$ we have:

$$h'(z) := [\![z \notin \mathcal{E}(x, q)]\!] h(z) + [\![z \in \mathcal{E}(x, q)]\!] p h(z)\,.$$

We shall now show that Equation (15) does indeed hold after the subroutine is called, which will complete the proof. To show this we consider each step of the subroutine in turn. After Step 2 we have (via induction) that:

- For all $v \in \Uparrow(q)$ we have $\phi(v) = 1$.

- For all $v \in \mathcal{D} \setminus \Uparrow(q)$ we have:

$$\psi(v) \prod_{v' \in \Uparrow(v)} \phi(v') = \sum_{z \in \Downarrow(v)} h(z)\,.$$

So, since $\mathcal{E}(x, q)$ is the set of all $z \in \mathcal{X}$ that do not lie to the left of q in $\mathcal{D}$ we have that, after Step 4 of the algorithm, the following holds:

- For all $v \in \Uparrow(q)$ we have $\phi(v) = 1$,
- For all $v \in \mathcal{D} \setminus \Uparrow(q)$ we have:

$$\psi(v) \prod_{v' \in \Uparrow(v)} \phi(v') = \sum_{z \in \Downarrow(v)} h'(z)\,.$$

Hence, by induction, we have that, after Step 5 of the algorithm, it is the case that for all $v \in \Uparrow(q)$ we have: $\psi(v) = \sum_{z \in \Downarrow(v)} h'(z)$. So since $\phi(v) = 1$ for all $v \in \Uparrow(q)$ and Step 5 does not alter $\phi(v)$ or $\psi(v)$ for any $v \in \mathcal{D} \setminus \Uparrow(q)$ we have Equation (15). ■

E Lower bound proof

Proposition E.1. *Take any learning algorithm. Given any basis $\mathcal{B}$ and any $M \in \mathbb{N}$ then for any sequence of disjoint basis elements $\langle B_j \mid j \in [M] \rangle$ there exists a sequence of corresponding actions $\langle b_j \in [K] \mid j \in [M] \rangle$ such that an adversary can force:*

$$\sum_{t \in [T]} \sum_{j \in [M]} [\![x_t \in \mathcal{B}_j]\!] r_{t,b_j} - \sum_{t \in [T]} \mathbb{E}[r_{t,a_t}] \in \Omega(\sqrt{MKT})$$

Proof. In this scenario, at each time step, either a single expert (i.e., the basis element containing the current context x_t) is active, making predictions based on its label, or no expert is active, prompting the learner to abstain and thus incur zero reward or cost.

Therefore we define $T' = \{t \in [T] \mid \sum_{j \in [M]} [\![x_t \in \mathcal{B}_j]\!] = 1\}$ as the set of timesteps in which the learner is going to play. Since the concept of abstention is that our algorithm is not going to pay anything for the timesteps in which we abstain, we can see that:

$$\sum_{t \in [T]} \sum_{j \in [M]} [\![x_t \in \mathcal{B}_j]\!] r_{t,b_j} - \sum_{t \in [T]} \mathbb{E}[r_{t,a_t}] = \sum_{t \in T'} r_{t,b_j} - \sum_{t \in T'} \mathbb{E}[r_{t,a_t}]\,,$$

For any ball $j \in [M]$, we define $T_j = \{t \in [T'] \mid [\![x_t \in \mathcal{B}_j]\!]\}$. Following the ideas of Seldin and Lugosi [2016], for any of the sets T_j we can create a multi-armed bandit instance as the one described in the lower bound by Auer et al. [2002]. Note that in the lower bound construction, the abstention arm would be a forehand known suboptimal arm, which results in a lower bound of the order $c\sqrt{(K-1)T}$, for the constant $c = \frac{\sqrt{2}-1}{\sqrt{32\ln(4/3)}} > 0$. Since the presented context x_t is chosen adversarially at each time step, we can ensure that each basis element is activated for $|T'|/M$ time steps, obtaining:

$$\begin{aligned}
\sum_{j \in [M]} \left(\sum_{s \in T'_j} r_{s,b_j} - \sum_{s \in T'_j} \mathbb{E}[r_{s,a_s}] \right) &\geq \sum_{j \in [M]} c\sqrt{(K-1)|T'_j|} \\
&= \sum_{j \in [M]} c\sqrt{(K-1)|T'|/M} \\
&= c\sqrt{M(K-1)|T'|}
\end{aligned}$$

As we can choose $|T'|$ to be any fraction of T, we end up with the desired lower bound of the order $\Omega(\sqrt{MKT})$, which matches, up to logarithmic factors, the cumulative reward bound presented in Theorem 5.3. □

F Overlapping balls extension

In this section, we present the theorem that allows us to present the results of overlapping balls as expressed in Section 5.2. Note that Theorem 5.3 is the special case of Theorem F.1 when the balls are disjoint and $u_j = 1$ for all $j \in [M]$.

Theorem F.1. *Let $M \in \mathbb{N}$ and $\{(B_j, b_j, u_j) \mid j \in [M]\}$ be any sequence such that B_j is a ball, $b_j \in [K]$ is an action, and $u_j \in [0,1]$ is such that for all $x \in \mathcal{X}$ we have:*

$$\sum_{j\in[M]} [\![x \in B_j]\!] u_j \leq 1\,.$$

For all $t \in [T]$ define:

$$r_t^* := \sum_{j\in[M]} [\![x_t \in B_j]\!] u_j r_{t,b_j}\,,$$

which represents the reward of the policy induced by $\{(B_j, b_j, u_j) \mid j \in [M]\}$ on trial t. The regret of CBA*, with the set of experts given in Section 5.2 and with correctly tuned parameters, is then bounded by:*

$$\sum_{t\in[T]} r_t^* - \sum_{t\in[T]} \mathbb{E}[r_{t,a_t}] \in \mathcal{O}\left(\sqrt{\ln(KN)KT\sum_{j\in[M]} u_j}\right)\,.$$

Its per-trial time complexity is:

$$\mathcal{O}(KN\ln(N))\,.$$

Proof. Direct from Theorem 3.1 using the experts (with efficient implementation) given in Section 5.2 $\square$

G The details of the graph bases

This section expands the definition and explanations for the bases we used in the Experiment. Remember that we refer to any set of experts that correspond to set-action pairs of the form $(B, k) \in 2^{\mathcal{X}} \times [K]$ as a *basis elements*, and a set of basis elements as *basis*.

G.1 p-seminorm balls on graphs

As we see in Sec. 5.2, the CBA seems to work only for vector data. However, in the following sections, we explore how our CBA algorithm can be applied to graph data by creating a ball structure over the graph.

We first introduce the notations of a graph. A graph is a pair of *nodes* $V := [N]$ and *edges* E. An edge connects two nodes, and we assume that our graph is *undirected* and *weighted*. For each edge $\{i, j\} \in E$, we denote its weight by c_{ij}. For convenience, for each pair of nodes i, j with $\{i, j\} \notin E$, we define $c_{ij} = 0$.

To form a ball over a graph, a family of metrics we are particularly interested in is given by p-norms on a given graph G. Let

$$d_p(i,j) := \left(\min_{\substack{\boldsymbol{u}\in\mathbb{R}^N \\ u_i - u_j = 1}} \sum_{s,t\in V} c_{st}|u_s - u_t|^p\right)^{-1/p}\,. \tag{16}$$

which is a well-defined metric for $p \in [1, \infty)$ if the graph is connected and may be defined for $p = \infty$ by taking the appropriate limits. When $p = 2$ this is the square root of the *effective resistance* circuit between nodes i and j which comes from interpreting the graph as an electric circuit where the edges are unit resistors and the denominator of Equation (16) is the power required to maintain a unit voltage difference between u and v [Doyle and Snell, 1984]. More generally, $d_p(i,j)^p$ is known as p-(effective) resistance [Herbster and Lever, 2009, Alamgir and von Luxburg, 2011, Saito and Herbster, 2023]. When $p \in \{1, 2, \infty\}$ there are natural interpretation of the p-resistance. In the case of $p = 1$, we have that the effective is equal to one over the number of edge-disjoint paths between i

and j which is equivalently one over the minimal cut that separates i from j. When $p = 2$ it is the effective resistance as discussed above. And finally when $p = \infty$ we have that d_∞ is the geodesic distance (shortest path) between i and j. Note that, interestingly, there are at most $2N$ distinct balls for d_1; as opposed to the general bound $O(N^2)$ on the number of metric balls. This follows since d_1 is an *ultrametric*. A nice feature of metric balls is that they are ordinal, i.e., we can take an increasing function of the distance and the distinct are unchanged. The time complexity for each ball is as follows. For d_1 ball, we compute every pair of distance in $\mathcal{O}(N^3)$ using the Gomory-Hu tree [Gomory and Hu, 1961]. For d_2 ball, it is actually enough to compute the pseudoinverse of graph Laplacian once, which costs $\mathcal{O}(N^3)$ [Doyle and Snell, 1984]. For d_∞ ball, we can compute every pair of distance in $\mathcal{O}(N^3)$ by Floyd–Warshall algorithm [Floyd, 1962].

G.2 Community detection bases

In this section, we consider only bases formed via a set of subsets (a.k.a clusters) $C \subseteq 2^{[N]}$. Each of these subsets induces K basis elements: one for each action $a \in [K]$. Specifically, the basis element $\beta : [N] \to [K_\Box]$ corresponding to the pair (C, a) is such that $\beta(x)$ is equal to a whenever $x \in C$ and equal to $\Box$ otherwise. Hence, in this section, we equate a basis with a set of subsets of $[N]$.

We can compute a basis for a given graph $G = (V, E)$ using community detection algorithms. Community detection is one of the most well-studied operations for graphs, where the goal is to find a partition $\{C_1, \dots, C_q\}$ of V (i.e., $\bigcup_{i=1}^q C_i = V$ and $C_i \cap C_j = \emptyset$ for $i \neq j$) so that each C_i is densely connected internally but sparsely connected to the rest of the graph [Fortunato, 2010]. There are many community detection algorithms, all of which can be used here, but the most popular algorithm is the Louvain method [Blondel et al., 2008]. We briefly describe how this algorithm works. The algorithm starts with an initial partition $\{\{v\} \mid v \in V\}$ and aggregates the clusters iteratively: For each $v \in V$, compute the gain when moving v from its current cluster to its neighbors' clusters and indeed move it to a cluster with the maximum gain (if the gain is positive). Note that the gain is evaluated using *modularity*, i.e., the most popular quality function for community detection [Newman and Girvan, 2004]. The algorithm repeats this process until no movement is possible. Then the algorithm aggregates each cluster to a single super node (with appropriate addition of self-loops and change of edge weights) and repeats the above process on the coarse graph as long as the coarse graph is updated. Finally, the algorithm outputs the partition of V in which each cluster corresponds to each super node in the latest coarse graph. Note that it is widely recognized that the Louvain method works in $\mathcal{O}(N \log N)$ in practice [Traag, 2015].

To obtain a finer-grained basis, we apply the so-called greedy peeling algorithm for each C_i in the output of the Louvain method. For $C_i \subseteq V$ and $v \in C_i$, we denote by $d_{C_i}(v)$ the degree of v in the induced subgraph $G[C_i]$. For $G[C_i]$, the greedy peeling iteratively removes a node with the smallest degree in the currently remaining graph and obtains a sequence of node subsets from C_i to a singleton. Specifically, it works as follows: Set $j \leftarrow |C_i|$ and $C_i^{(j)} \leftarrow C_i$. For each $j = |C_i|, \dots, 2$, compute $v_{\min} \in \arg\min\{d_{C_i^{(j)}}(v) \mid v \in C_i^{(j)}\}$ and $C_i^{(j-1)} \leftarrow C_i^{(j)} \setminus \{v_{\min}\}$. Using a sophisticated data structure, this algorithm runs in linear time [Lanciano et al., 2024].

In summary, our community detection basis is the collection of node subsets $\{C_i^{(j)} \mid i = 1, \dots, q,\ j = 1, \dots, |C_i|\}$ together with $\{\{v\} \mid v \in V\}$ for completeness.

G.3 Graph convexity bases

An alternative to metric balls and communities are, for example, (geodesically) convex sets in a graph. They correspond to the inductive bias that if two nodes prefer the same action, then also the nodes on a shortest path between the two tend to prefer the same action. Geodesically convex sets are well-studied [van De Vel, 1993, Pelayo, 2013] and have been recently used in various learning settings on graphs [Bressan et al., 2021, Thiessen and Gärtner, 2021]. Similarly to convex sets in the Euclidean space, a set C of nodes is *convex* if the nodes of any shortest path with endpoints in C are in C, as well. More formally, the (geodesic) *interval* $I(u, v) = \{x \in V : x \text{ is on a shortest path between } u \text{ and } v\}$ of two nodes u and v contains all the nodes on a shortest path between them. For a set of node A we define $I(A) = \cup_{a,b \in A} I(a, b)$ as a shorthand notation for the union of all pairwise intervals in A. A set A is (geodesically) convex iff $I(A) = A$ and the *convex hull* $\mathrm{conv}(A)$ of a set A is the (unique) smallest convex set containing A. Note that for $u, v \in V$, $I(u, v)$ and $\mathrm{conv}(\{u, v\})$ are typically different sets. Indeed, $I(u, v)$ is in general non-convex, as nodes on a shortest path between two nodes in $I(u, v)$ (except for u, v) are not necessarily contained in $I(u, v)$. As the total number of convex sets can be exponential in N, e.g., all subsets of a complete subgraph are convex, we consider the basis consisting of all intervals: $I(u, v)$ for $u, v \in [N]$. This involves $\mathcal{O}(N^2)$ basis elements, each of size $\mathcal{O}(N)$. With a simple modification of the Floyd Warshall [Floyd, 1962] algorithm, computing the interval basis takes $\mathcal{O}(N^3)$ time complexity.

H Additional experimental results

We thoroughly explored various configurations for the three graphs described in our experimental setup in Section 6. We run our experiments with an Intel Xeon Gold 6312U processor and 256 GB of RAM ECC 3200 MHz. Figure 3 displays different settings for the number of nodes in each clique and noise levels.

As we compare the computational complexity of each basis in Section G and the main results, the most intense computational load in the experiments will arise from the calculation of the basis, which can be seen as an initialization step in our algorithm. The proposed methods have varying computational complexities, and an arbitrarily complex function can be employed to compute the basis. Remark that, in the usual complexity comparison among online learning algorithms using experts, we compare the complexity *given* the experts. Practically, we use pre-computed bases or even human experts. Also note that due to the expensive complexity of the p-balls and the convex sets seen in Section G, we only conduct the LVC for LastFM Asia.

In Figure 4, we present multiple settings for generating the Gaussian graph. Here the title of each plot is "Foreground x,y; Background x',y'; k-NN," which is explained as follows: x represents the number of nodes in each foreground class, x' represents the number of nodes in the background class, y represents the standard deviation of the Gaussians generating the foreground class, y' represents the standard deviation of the Gaussian generating the background class, and k represents the number of nearest neighbors used to generate the graph.

In Figure 5, we present the various labels chosen as noise for the Cora graph. In Figure 2(c), we presented the averages of all these different configurations. Here, we can see that the main behavior of the various bases is roughly maintained independently of the different labels chosen to be masked as background class.

In Figure 6, we present the various labels chosen as noise for the LastFM Asia graph. This graph comprises nodes representing LastFM users in Asian countries and edges representing mutual follower connections. Vertex features are extracted based on the artists liked by the users. During this initial analysis, we arbitrarily chose three out of eighteen possible labels to serve as the background class. In Figure 2(d), we presented the averages of all these different configurations. Varying the chosen background classes also produces different results, this is indeed due to the inherent lack of noise in the dataset. It is nice to see that regardless of the noise labels chosen, the behavior of our algorithm is always good, showing, as expected, that based on the amount of noise, we can just improve.

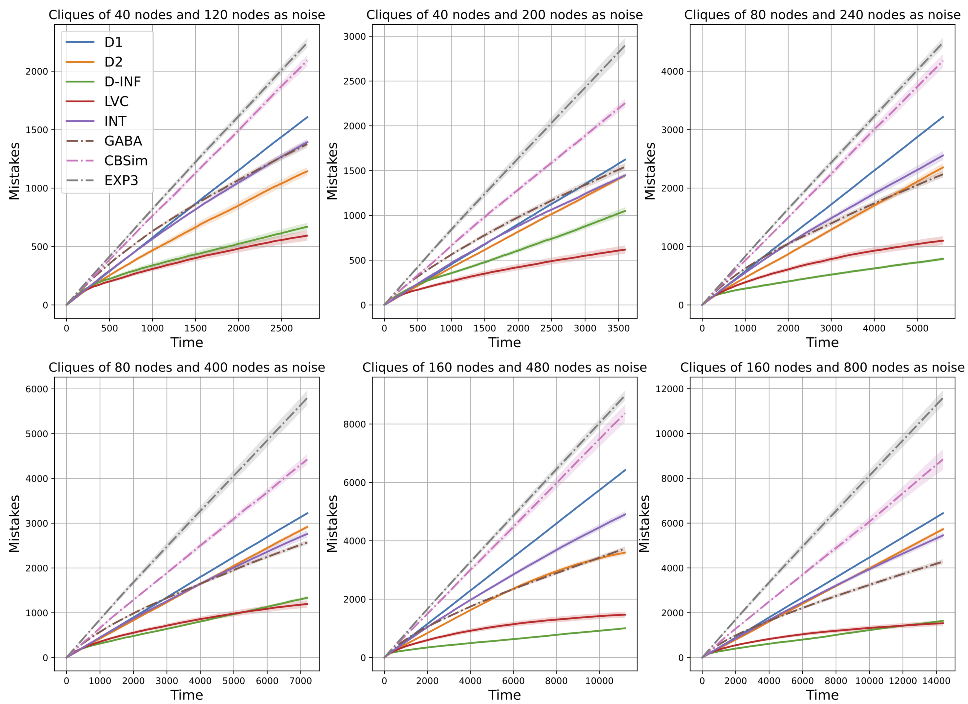


Figure 3: Stochastic Block Model results, dotted lines represent different baselines, while solid lines are used to represent various results.

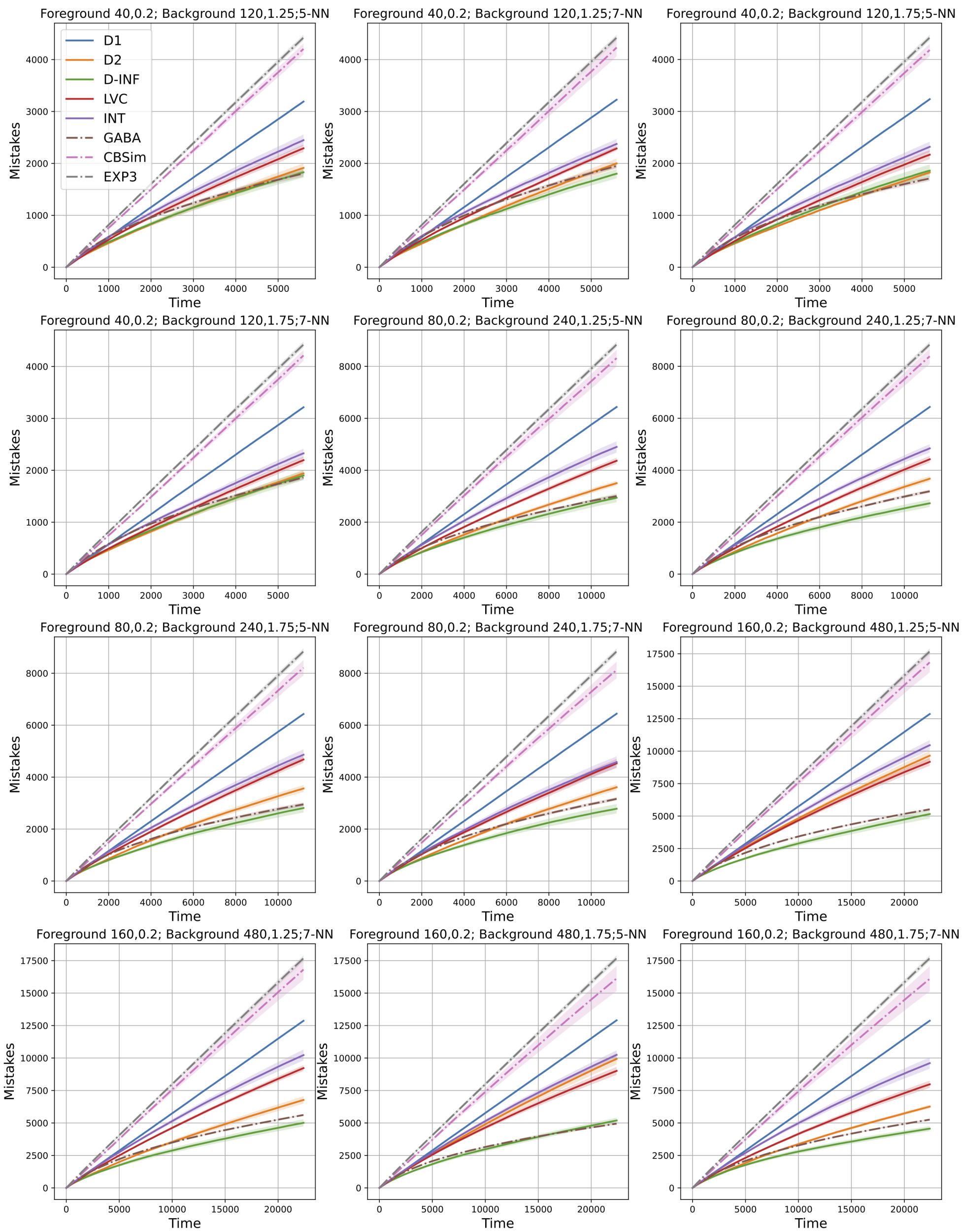


Figure 4: Gaussian graph results, dotted lines represent different baselines, while solid lines are used to represent various results.

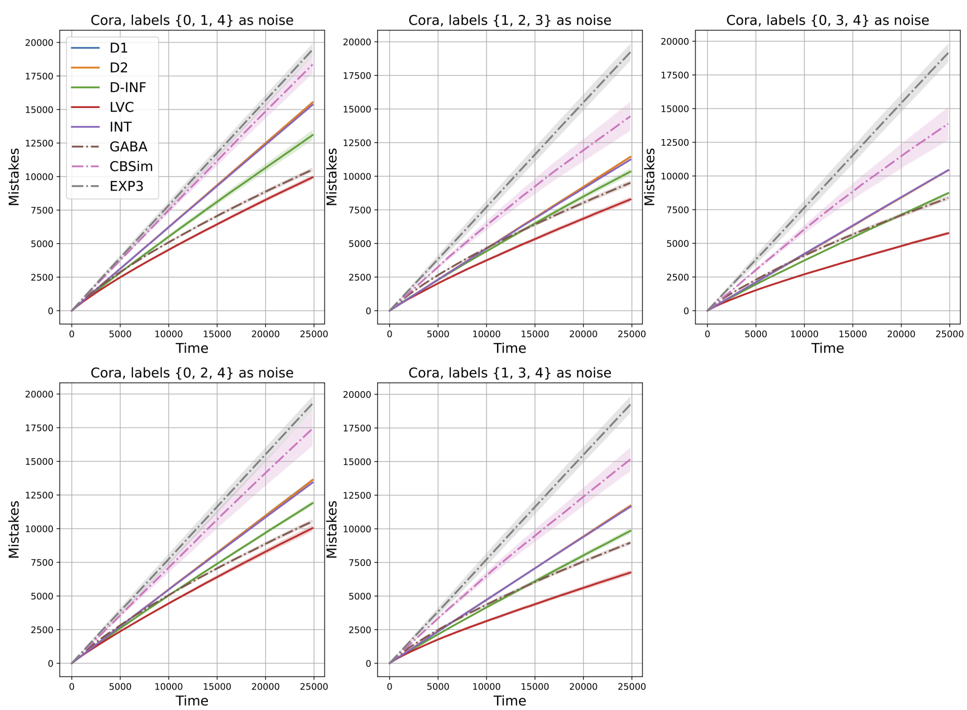


Figure 5: Cora results, dotted lines represent different baselines, while solid lines are used to represent various results

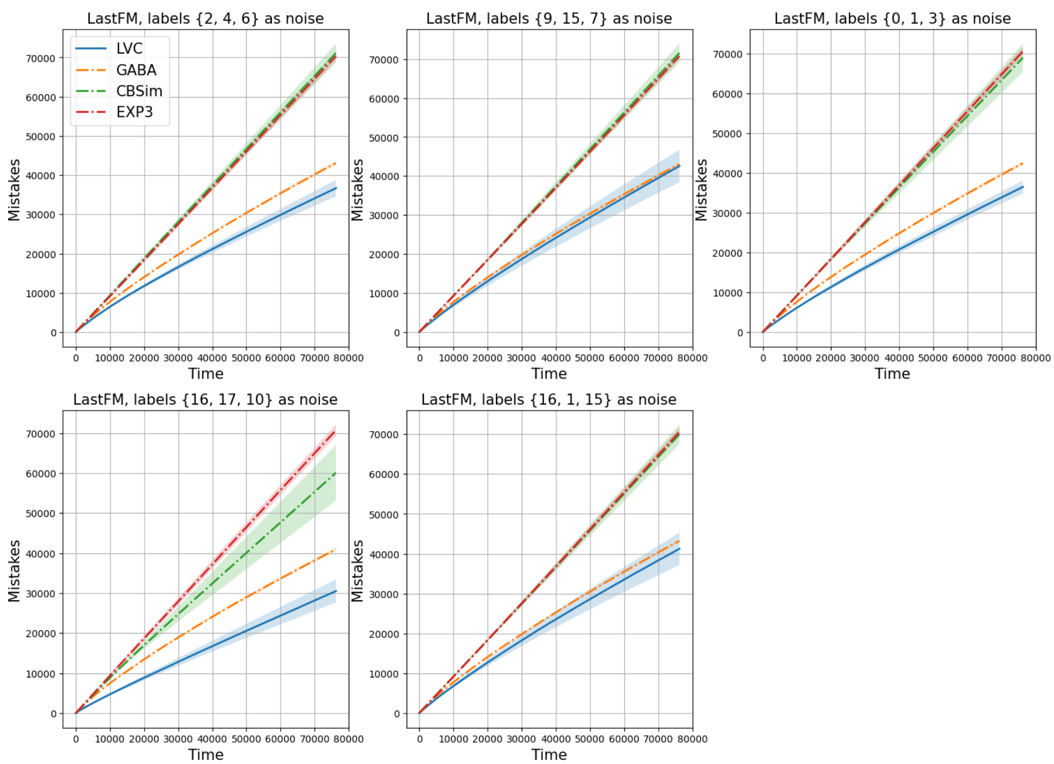


Figure 6: LastFM Asia results, dotted lines represent different baselines, while solid lines are used to represent various results