Practical Evaluation and Optimization of Contextual Bandit Algorithms

Alberto Bietti, Alekh Agarwal, John Langford

Practical Evaluation and Optimization of Contextual Bandit Algorithms

Alberto Bietti
Inria, MSR-Inria Joint Center
alberto.bietti@inria.fr

Alekh Agarwal
Microsoft Research NY
alekha@microsoft.com

John Langford
Microsoft Research NY
jcl@microsoft.com

February 12, 2018

Abstract

We study and empirically optimize contextual bandit learning, exploration, and problem encodings across 500+ datasets, creating a reference for practitioners and discovering or reinforcing a number of natural open problems for researchers. Across these experiments we show that minimizing the amount of exploration is a key design goal for practical performance. Remarkably, many problems can be solved purely via the implicit exploration imposed by the diversity of contexts. For practitioners, we introduce a number of practical improvements to common exploration algorithms including Bootstrap Thompson sampling, Online Cover, and ϵ-greedy. We also detail a new form of reduction to regression for learning from exploration data. Overall, this is a thorough study and review of contextual bandit methodology.

1 Introduction

At a practical level, how should contextual bandit learning and exploration be done?

In the contextual bandit problem, a learner repeatedly observes a context, chooses an action, and observes a loss for the chosen action only. Many real-world interactive machine learning tasks are well-suited for this setting: a movie recommendation system selects a movie for a given user and receives feedback (click or no click) only for that movie; a choice of medical treatment may be prescribed to a patient with an outcome observed for (only) the chosen treatment. The limited feedback (known as *bandit* feedback) received by the learner highlights the importance of *exploration*, which needs to be addressed by contextual bandit algorithms.

Contextual bandit learning naturally decomposes into several elements, each of which can be optimized.

1. What is the best *encoding* of the value of feedback? In particular, if values have a range of 1, should we encode the costs of best and worst outcomes as 0/1, -1/0, or 9/10? Although some techniques (Dudik et al., 2011b) attempt to remove a dependence on encoding, in practice, and particularly in an online scenario, this is imperfect.

2. Many successful exploration algorithms learn by *reduction* to supervised learning, in an off-policy scenario from a batch of exploration data. What is the best reduction to use in this context? Much of the theory depends on inverse propensity scoring, but more powerful estimators (*e.g.*, Dudik et al., 2011b) can be used.

3. What is the best exploration algorithm? The focal point of contextual bandit learning research is efficient exploration (Agarwal et al., 2012, 2014; Agrawal and Goyal, 2013; Dudik et al., 2011a; Langford and Zhang, 2008; Russo et al., 2017). However, many of these algorithms remain far from practical, and even when considering more practical variants, their empirical behavior is poorly understood.

The primary goal of this paper is to better understand the empirical behavior of various practical contextual bandit algorithms. To this end, we evaluate these on a collection of over 500 multiclass classification datasets,

Figure 1: A greedy approach can outperform an optimized exploration method in many cases. Red points indicate datasets where there is a significant difference in loss between the two methods.

which simulate contextual bandit problems by revealing the loss of only one chosen label to the learner. All our experiments are based on the online learning system Vowpal Wabbit[1] which has already been successfully used in production systems (Agarwal et al., 2016).

A secondary goal of this paper is fixing issues displayed by existing exploration algorithms so as to maximize practical performance. For this purpose, we propose several algorithmic alternatives.

1. For basic online learning, it turns out that learning a feature-independent prediction and then regressing on the residual is remarkably helpful to make learning robust to loss encodings. See Section 2.4.

2. When reducing contextual bandit learning to supervised learning, a technique we call 'importance-weighted regression' is provably unbiased while offering both computational and statistical benefits. See Section 2.3.

3. When doing basic ϵ-greedy style exploration, it is natural to restrict exploration to only those actions which plausibly could be taken by the best policy. Using techniques from active learning (*e.g.*, Hsu, 2010), we can efficiently discover and randomize over just those actions, achieving a regret guarantee of $O(T^{1/3})$ in favorable settings. See Section 3.4.

4. For more efficient methods to do exploration, such as the cover approach (Agarwal et al., 2014) or a Bootstrap version of Thompson sampling, we propose improvements to previous implementations that result in superior empirical performance. See Sections 3.2 and 3.3.

Remarkably, we discover that even after all of the above tweaks doing *no* explicit exploration whatsoever is often the best approach! Figure 1 shows that such a greedy approach often outperformed an optimized exploration method in our study. This lends strong empirical credence to approaches seeking a good regret guarantee when there is sufficient diversity in the data (Bastani et al., 2017; Kannan et al., 2018) as well as open problems related to improving data-dependent guarantees (Agarwal et al., 2017).

In summary, we provide a comprehensive evaluation of contextual bandit algorithms and improvements to several algorithmic aspects based on our learnings.

2 Contextual Bandit Setup

In this section, we present the learning setup considered in this paper, recalling the stochastic contextual bandit setting, the notion of optimization oracle, various techniques used by contextual bandit algorithms for leveraging these oracles, and finally our experimental setup.

[1] `http://hunch.net/~vw/`

2.1 Learning Setting

The stochastic (i.i.d.) contextual bandit learning problem can be described as follows. At each time step t, the environment produces a pair $(x_t, \ell_t) \sim D$, where $x_t \in \mathcal{X}$ is a context vector and $\ell_t = (\ell_t(1), \ldots, \ell_t(K)) \in \mathbb{R}^K$ is a loss vector, with K the number of possible actions. After observing the context x_t, the learner chooses an action a_t, and only observes the loss $\ell_t(a_t)$ corresponding to the chosen action. The goal of the learner is to trade-off exploration and exploitation in order to incur a small cumulative regret

$$R_T := \sum_{t=1}^{T} \ell_t(a_t) - \min_{\pi \in \Pi} \sum_{t=1}^{T} \ell_t(\pi(x_t)),$$

where Π is a (large, and possibly infinite) set of policies $\pi : \mathcal{X} \to \{1, \ldots, K\}$. It is often important for the learner to use randomized strategies, hence we let $p_t(a) \in [0, 1]$ denote the probability that the agent chooses action $a \in \{1, \ldots, K\}$, so that $a_t \sim p_t$.

2.2 Optimization Oracle

In this paper, we focus on contextual bandit algorithms which rely on access to an *optimization oracle* for solving optimization problems similar to those that arise in supervised learning, leading to methods that are suitable for general policy classes Π. The main example is the *cost-sensitive classification* (CSC) oracle (see, *e.g.*, Agarwal et al., 2014; Dudik et al., 2011a; Langford and Zhang, 2008), which given a collection $(x_1, c_1), \ldots, (x_T, c_T) \in \mathcal{X} \times \mathbb{R}^K$ computes

$$\arg\min_{\pi \in \Pi} \sum_{t=1}^{T} c_t(\pi(x_t)). \tag{1}$$

The cost vectors $c_t \in \mathbb{R}^K$ are often constructed using counterfactual estimates of the true (unobserved) losses, as we describe below. Other work (*e.g.*, Agarwal et al., 2012) has considered the use of *regression* oracles which compute $\arg\min_{f \in \mathcal{F}} \sum_{t=1}^{T} (f(x_t, a_t) - y_t)^2$, where $\mathcal{F}$ is a class of functions $f : \mathcal{X} \times \{1, \ldots, K\} \to \mathbb{R}$ that aim to predict a cost y_t from a given context x_t and action a_t. In this paper, we also consider the following more general regression oracle with importance weights $\omega_t > 0$:

$$\arg\min_{f \in \mathcal{F}} \sum_{t=1}^{T} \omega_t (f(x_t, a_t) - y_t)^2. \tag{2}$$

While the theory typically requires exact solutions to (1) or (2), this is often impractical due to the difficulty of the underlying problem (especially for cost-sensitive classification), and more importantly because the size of the optimization problems to be solved keeps increasing after each iteration. In this work, we consider instead the use of *online* optimization for solving problems (1) or (2), by incrementally updating a given policy or regression function after each new observation, using for instance an online gradient method, which is natural in an online learning context such as the contextual bandit setting.

2.3 Loss Estimates and Reductions

A common approach to solving problems with bandit (partial) feedback is to compute an estimate of the full feedback using the observed loss and then apply methods for the full-information setting to these estimated values. In the case of contextual bandits, these loss estimates are commonly used to create cost-sensitive classification instances to be solved by the optimization oracle introduced above (Agarwal et al., 2014; Dudik et al., 2011a; Langford and Zhang, 2008). This is sometimes referred as *reduction* to cost-sensitive classification. We now describe the three different estimation methods considered in this paper, and how each is typically used for reduction to an optimization oracle. In what follows, we consider an observed interaction record $(x_t, a_t, \ell_t(a_t), p_t(a_t))$.

Perhaps the simplest approach is the **inverse propensity-scoring** (IPS) estimator:

$$\hat{\ell}_t(a) := \frac{\ell_t(a_t)}{p_t(a_t)} \mathbb{1}\{a = a_t\}. \tag{3}$$

For any action a in the support of p_t ($p_t(a) > 0$), this estimator is unbiased, *i.e.* $\mathbb{E}_{a_t \sim p_t}[\hat{\ell}_t(a)] = \ell_t(a)$, but can have high variance when $p_t(a_t)$ is small. This leads to a straightforward cost-sensitive classification example $(x_t, \hat{\ell}_t)$. Using such examples in (1) provides a way to perform off-policy (or counterfactual) evaluation and optimization, which in turn allows a contextual bandit algorithm to identify good policies for exploration. In order to obtain good unbiased estimates, one needs to control the variance of the estimates, *e.g.*, by enforcing a minimum exploration probability $p_t(a) \geq \epsilon > 0$ on all actions.

In order to reduce the variance of IPS, the **doubly robust** (DR) estimator (Dudik et al., 2011b) uses a separate, possibly biased, estimator of the loss $\hat{\ell}(x, a)$:

$$\hat{\ell}_t(a) := \frac{\ell_t(a_t) - \hat{\ell}(x_t, a_t)}{p_t(a_t)} \mathbb{1}\{a = a_t\} + \hat{\ell}(x_t, a). \tag{4}$$

When $\hat{\ell}(x_t, a_t)$ is a good estimate of $\ell_t(a_t)$, the small numerator in the first term helps reduce the variance induced by a small denominator, while the second term ensures that the estimator is unbiased. Typically, $\hat{\ell}(x, a)$ is learned by regression on all past observed losses. The reduction to cost-sensitive classification is similar to IPS.

We introduce a third method that directly reduces to the importance-weighted regression oracle (2), which we call IWR (for **importance-weighted regression**). This approach finds a regression function

$$\hat{f} := \arg\min_{f \in \mathcal{F}} \sum_{t=1}^{T} \frac{1}{p_t(a_t)} (f(x_t, a_t) - \ell_t(a_t))^2, \tag{5}$$

and considers the policy $\hat{\pi}(x) = \arg\min_a \hat{f}(x, a)$. Note that if p_t has full support, then the objective is an unbiased estimate of the full regression objective on all actions

$$\sum_{t=1}^{T} \sum_{a=1}^{K} (f(x_t, a) - \ell_t(a))^2.$$

In contrast, if the learner only explores a single action (so that $p_t(a_t) = 1$ for all t), and if we consider a linear class of regressors of the form $f(x, a) = \theta_a^\top x$ with $x \in \mathbb{R}^d$, then the IWR reduction computes least-squares estimates $\hat{\theta}_a$ from the data observed when action a was chosen. When actions are selected according to the greedy policy $a_t = \arg\min_a \hat{\theta}_a^\top x_t$, this setup corresponds to the greedy algorithm considered, *e.g.*, in (Bastani et al., 2017).

Note that while CSC is typically intractable and requires approximations in order to work in practice, importance-weighted regression does not suffer from these issues, though a general theoretical analysis is lacking. In addition, while the computational cost for an approximate CSC online update scales with the number of actions K, IWR only requires an update for a single action, making the approach more attractive computationally. Another benefit of IWR in an online setting is that it can leverage importance weight aware online updates (Karampatziakis and Langford, 2011), which makes it easier to handle large inverse propensity scores.

2.4 Experimental Setup

Our experiments are conducted by simulating the contextual bandit setting using multiclass classification datasets, and use the online learning system Vowpal Wabbit (VW).

Algorithm 1 Generic contextual bandit algorithm

for $t = 1, \dots$ **do**
  Observe context x_t, compute $p_t = \texttt{explore}(x_t)$;
  Choose action $a_t \sim p_t$, observe loss $\ell_t(a_t)$;
  $\texttt{learn}(x_t, a_t, \ell_t(a_t), p_t)$;
end for

Simulated contextual bandit setting. The experiments in this paper are based on conversion from multiclass classification to contextual bandit learning. In particular, we convert a multiclass classification example $(x_t, y_t) \in \mathcal{X} \times \{1, \dots, K\}$ into an contextual bandit example $(x_t, \ell_t) \in \mathcal{X} \times \mathbb{R}^K$ such that $\ell_t(y_t) < \ell_t(a)$ for $a \neq y_t$, and we only reveal the loss $\ell_t(a_t)$ of the action a_t chosen by the contextual bandit algorithm. We consider losses defined as:

$$\ell_t^c(a) = c + \mathbb{1}\{a \neq y_t\}, \tag{6}$$

for some $c \in \mathbb{R}$, which is taken in $\{0, -1, 9\}$ in our experiments. The behavior observed for different choices of c allows us to get a sense of the robustness of the algorithms to the scale of observed losses, which might be unknown. Separately, different values of c can lead to low estimation variance in different scenarios: $c = 0$ might be preferred if $\ell_t^0(a)$ is often 0, while $c = -1$ is preferred when $\ell_t^0(a)$ is often 1. In order to have a meaningful comparison between different algorithms, loss encodings, as well as supervised multiclass classification, we evaluate using ℓ_t^0 with progressive validation (Blum et al., 1999).

Online learning in VW. Online learning is an important tool for having machine learning systems that quickly and efficiently adapt to observed data (Agarwal et al., 2016; He et al., 2014; McMahan et al., 2013). We run our contextual bandit algorithms in an online fashion using Vowpal Wabbit: instead of exact solutions of the optimization oracles from Section 2.2, we consider online CSC or regression oracles. Online CSC itself reduces to multiple online regression problems in VW, and online regression updates are performed using adaptive (Duchi et al., 2011), normalized (Ross et al., 2013) and importance-weight-aware (Karampatziakis and Langford, 2011) gradient updates, with a single tunable step-size parameter.

Parameterization and *baseline*. We consider linearly parameterized policies of the form $\pi(x) = \arg\min_a \theta_a^\top x$, or in the case of the IWR reduction, regressors $f(x, a) = \theta_a^\top x$. For the DR loss estimator, we use a similar linear parameterization $\hat{\ell}(x, a) = \phi_a^\top x$. We also consider the use of an *action-independent additive baseline* term in our loss estimators, which can help learn better estimates with fewer samples. In this case the regressors take the form $f(x, a) = \theta_0 + \theta_a^\top x$ (IWR) or $\hat{\ell}(x, a) = \phi_0 + \phi_a^\top x$ (DR). In order to learn the baseline term more quickly, we propose to use a separate online update for the parameters θ_0 or ϕ_0 to regress on observed losses, followed by an online update on the residual for the action-dependent part. We scale the step-size of these baseline updates by the largest observed magnitude of the loss, in order to adapt to the observed loss range in the case of normalized online updates (Ross et al., 2013).

3 Algorithms

In this section, we present the main algorithms we study in this paper, namely *ϵ-greedy*, *bagging* and *cover*, and modifications that can achieve reduced exploration. All methods are based on the generic scheme in Algorithm 1. `explore` computes the exploration distribution p_t over actions, and `learn` updates the algorithm's policies. For simiplicity, we consider a function `oracle` which performs an online update to a policy using IPS, DR or IWR reductions given an interaction record. In some cases, the CSC oracle is called explicitly with different cost vectors, and we denote such a call by `csc_oracle`, along with a loss estimator `estimator` which takes an interaction record and computes IPS or DR loss estimates.

Algorithm 2 ϵ-greedy

π_1; $\epsilon > 0$ (or $\epsilon = 0$ for Greedy).
`explore`(x_t):
  return $p_t(a) = \epsilon/K + (1-\epsilon)\,\mathbb{1}\{\pi_t(x_t) = a\}$;
`learn`$(x_t, a_t, \ell_t(a_t), p_t)$:
  $\pi_{t+1} =$ `oracle`$(\pi_t, x_t, a_t, \ell_t(a_t), p_t(a_t))$;

Algorithm 3 Bag

$\pi_1^1, \ldots, \pi_1^N$.
`explore`(x_t):
  return $p_t(a) \propto |\{i : \pi_t^i(x_t) = a\}|$;
`learn`$(x_t, a_t, \ell_t(a_t), p_t)$:
  for $i = 1, \ldots, N$ **do**
    $\tau^i \sim Poisson(1)$; {with $\tau^1 = 1$ for bag-greedy}
    $\pi_{t+1}^i =$ `oracle`$^{\tau^i}(\pi_t^i, x_t, a_t, \ell_t(a_t), p_t(a_t))$;
  end for

3.1 ϵ-greedy and greedy

ϵ-greedy. We consider an importance-weighted variant of the epoch-greedy approach of Langford and Zhang (2008), given in Algorithm 2.

Greedy. When taking $\epsilon = 0$, only the greedy action given by the current policy is explored. With the IWR reduction and the linear regressors described in Section 2.4, this corresponds to an online version of the greedy algorithm in (Bastani et al., 2017). Because our online CSC oracle is based on regression, the DR version is similar, but includes additional regression samples on unobserved actions, with current loss estimates as targets. Somewhat surprisingly, our experiments show that this greedy algorithm can perform very well in practice, despite being exploration-free (see Section 4).

3.2 Bagging

This approach, shown in Algorithm 3, maintains a collection of N policies $\pi_t^1, \ldots, \pi_t^N$ meant to approximate a posterior distribution over policies (or, in Bayesian terminology, the parameter that generated the data) via the Bootstrap. This approximate posterior is used to choose actions in a Thompson sampling fashion (Agrawal and Goyal, 2013; Chapelle and Li, 2011; Russo et al., 2017; Thompson, 1933). Each policy is trained on a different online bootstrap sample of the observed data (Qin et al., 2013; Oza and Russell, 2001). The online bootstrap performs a random number τ of online updates to each policy instead of one (this is denoted by `oracle`$^\tau$). This is also known as online Bootstrap Thompson sampling (Eckles and Kaptein, 2014; Osband and Van Roy, 2015). In contrast to these works, which simply play the arm given by one of the N policies chosen at random, we compute the full action distribution p_t resulting from such a sampling, and leverage this information for improved importance weights in loss estimation as in previous work (Agarwal et al., 2014).

Greedy bagging. With a single policy ($N = 1$), Algorithm 3 resembles the Greedy algorithm, up to the randomized number of online updates. We found this method to often be outperformed by Greedy, thus our experiments consider a simple optimization of bagging where the first policy is always updated once ($\tau^1 = 1$ in Algorithm 3), which typically performs better than bagging, though not significantly for larger values of N.

Algorithm 4 Cover

$\pi_1^1, \ldots, \pi_1^N$; $\epsilon_t = \min(1/K, 1/\sqrt{Kt})$; $\psi > 0$.

`explore`(x_t):

  $p_t(a) \propto |\{i : \pi_t^i(x_t) = a\}|$;

  return $\epsilon_t + (1 - \epsilon_t)p_t$; {for cover}

  return p_t; {for cover-nu}

`learn`$(x_t, a_t, \ell_t(a_t), p_t)$:

  $\pi_{t+1}^1 =$ `oracle`$(\pi_t^1, x_t, a_t, \ell_t(a_t), p_t(a_t))$;

  $\hat{\ell}_t =$ `estimator`$(x_t, a_t, \ell_t(a_t), p_t(a_t))$;

  for $i = 2, \ldots, N$ **do**

    $q_i(a) \propto |\{j \leq i - 1 : \pi_{t+1}^j(x_t) = a\}|$;

    $\hat{c}(a) = \hat{\ell}_t(a) - \frac{\psi\epsilon_t}{\epsilon_t + (1-\epsilon_t)q_i(a)}$;

    $\pi_{t+1}^i =$ `csc_oracle`$(\pi_t^i, x_t, \hat{c})$;

  end for

3.3 Cover

This method, given in Algorithm 4, is based on Online Cover, an online approximation of the ILOVETOCONBANDITS algorithm of Agarwal et al. (2014). The approach maintains a collection of N policies, $\pi_t^1, \ldots, \pi_t^N$, meant to approximate a covering distribution over policies that are good for both exploration and exploitation. The first policy π_t^1 is trained on observed data using the oracle as in previous algorithms, while subsequent policies are trained using cost-sensitive examples which encourage diversity in the predicted actions compared to the previous policies.

Our implementation differs from the Online Cover algorithm of Agarwal et al. (2014, Algorithm 5) in how the diversity term in the definition of $\hat{c}(a)$ is handled (the second term). When creating cost-sensitive examples for a given policy π^i, this term rewards an action a that is not well-covered by previous policies (*i.e.*, small $q_i(a)$), by subtracting a term that decreases with $q_i(a)$ from the loss. While Online Cover considers a fixed $\epsilon_t = \epsilon$, we let ϵ_t decay with t, and introduce a parameter ψ to control the overall reward term, which bears more similarity with the analyzed algorithm. In particular, the magnitude of the reward is ψ whenever action a is not covered by previous policies (*i.e.*, $q_i(a) = 0$), but decays with $\psi\epsilon_t$ whenever $q_i(a) > 0$, so that the level of induced diversity can decrease over time as we gain confidence that good policies are covered.

Cover-NU. In order to reduce the level of exploration of Cover and be more competitive with the Greedy method, we propose a variant of Cover with no exploration outside of the actions chosen by covering policies, denoted by Cover-NU (for No Uniform exploration).

3.4 Active ϵ-greedy

The simplicity of the ϵ-greedy method described in Section 3.1 often makes it the method of choice for practitioners. However, the uniform exploration over randomly selected actions can be quite inefficient and costly in practice. A natural consideration is to restrict this randomization over actions which could plausibly be selected by the optimal policy $\pi^* = \arg\min_{\pi \in \Pi} L(\pi)$, where $L(\pi) = \mathbb{E}_{(x,\ell)\sim D}[\ell(\pi(x))]$ is the expected loss of a policy π.

To achieve this, we use techniques from disagreement-based active learning (Hanneke, 2014; Hsu, 2010). After observing a context x_t, for any action a, if we can find a policy π that would choose this action ($\pi(x_t) = a$) instead of the empirically best action $\pi_t(x_t)$, while achieving a small loss on past data, then there is disagreement about how good such an action is, and we allow exploring it. Otherwise, we are confident that the best policy would not choose this action, thus we avoid exploring it, and assign it a high cost. The resulting method is in Algorithm 5. For simplicity, we assume a $[0, 1]$ loss range (the function `to_01` denotes shifting and rescaling observed losses to $[0, 1]$, assuming their range is known), and assign a loss of 1 to

Algorithm 5 Active ϵ-greedy

π_1; ϵ; $C_0 > 0$.

`explore`(x_t):

$A_t = \{a : \texttt{loss\_diff}(\pi_t, x_t, a) \leq \Delta_{t,C_0}\}$;

$p_t(a) = \frac{\epsilon}{K} \mathbb{1}\{a \in A_t\} + (1 - \frac{\epsilon|A_t|}{K}) \mathbb{1}\{\pi_t(x_t) = a\}$;

return p_t;

`learn`$(x_t, a_t, \ell_t(a_t), p_t)$:

$\hat{\ell}_t = \texttt{estimator}(x_t, a_t, \texttt{to\_01}(\ell_t(a_t)), p_t(a_t))$;

$$\hat{c}_t(a) = \begin{cases} \hat{\ell}_t(a), & \text{if } p_t(a) > 0 \\ 1, & \text{otherwise.} \end{cases}$$

$\pi_{t+1} = \texttt{csc\_oracle}(\pi_t^i, x_t, \hat{c}_t)$;

such unexplored actions. The disagreement test we use is based on empirical loss differences, similar to the Oracular CAL active learning method (Hsu, 2010), denoted `loss_diff`, together with a threshold:

$$\Delta_{t,C_0} = \sqrt{C_0 \frac{K \log t}{\epsilon t}} + C_0 \frac{K \log t}{\epsilon t}.$$

A practical implementation of `loss_diff` for an online setting is given in Appendix A.1. We analyze a theoretical form of this algorithm in Appendix A.2, showing a formal version of the following theorem:

Theorem 1. *With high-probability, and under favorable conditions on disagreement and on the problem noise, active ϵ-greedy achieves expected regret $O(T^{1/3})$.*

Note that this data-dependent guarantee improves on worst-case guarantees achieved by the optimal algorithms of (Agarwal et al., 2014; Dudik et al., 2011a). In the extreme case where the loss of any suboptimal policy is bounded away from that of π^*, we show that our algorithm can achieve constant regret. While active learning algorithms suggest that data-dependent thresholds Δ_t can yield better guarantees (*e.g.*, Huang et al., 2015), this may require more work in our setting due to open problems related to data-dependent guarantees for contextual bandits (Agarwal et al., 2017). In a worst-case scenario, active ϵ-greedy behaves similarly to ϵ-greedy (Langford and Zhang, 2008), achieving an $O(T^{2/3})$ expected regret with high probability.

4 Evaluation

In this section, we present our evaluation of the contextual bandit algorithms described in Section 3.

4.1 Evaluation Methodology

Our evaluation consists or simulating a contextual bandit setting from multiclass classification datasets, as described in Section 2.4. This section describes the datasets, algorithms, and metrics used. All code will be made public.

Datasets. We consider a collection of 524 multiclass classification datasets from the `openml.org` platform, including among others, medical, gene expression, text, sensory or synthetic data. The precise list of datasets is in Appendix B, and Table 1 shows some statistics. Because of the online setup, we consider a fixed, shuffled ordering of each dataset.

Choices of algorithms. We evaluate the algorithms described in Section 3. We ran each method on every dataset with the following hyperparameters:

actions	#
2	403
3-9	73
10+	48

examples	#
$\leq 10^2$	94
10^2-10^3	270
10^3-10^5	132
$> 10^5$	28

features	#
≤ 50	392
51-100	35
101-1000	16
1000+	81

Table 1: Statistics on number of datasets by number of examples, actions and unique features in our collection of 524 datasets.

Method	Hyperparameters	Reduction	Encoding
G	-	IWR/DR	-1/0 + b
C-u/nu	$N \in \{4, 8, \mathbf{16}\}$ $\psi \in \{0.01, \mathbf{0.1}, 1\}$	DR	-1/0
B/B-g	$N \in \{4, 8, \mathbf{16}\}$	IWR	0/1 + b
ϵG	$\epsilon \in \{\mathbf{0.02}, 0.05, 0.1\}$	IWR	-1/0
A	$\epsilon \in \{0.05, \mathbf{1}\}$ $C_0 \in 10^{-\{2,4,\mathbf{6}\}}$	DR	0/1

Table 2: Choices of hyperparameters, and fixed choices of reduction and encoding for each method (G: greedy, C: cover, B: bag, A: active ϵ-greedy). Fixed choices of hyperparameters are in bold.

- algorithm-specific *hyperparameters*, shown in Table 2
- 9 choices of *learning rates*, on a logarithmic grid from 0.001 to 10 (see Section 2.4)
- 3 choices of loss *encodings*: 0/1, -1/0 and 9/10 (see Eq. (6))
- with or without *baseline* (denoted 'b', see Section 2.4)
- 3 choices of *reductions*: IPS, DR and IWR (see Section 2.3).

Unless otherwise specified, these are fixed to the choices highlighted in Table 2, which are chosen to optimize performance, except for the learning rate, which is always optimized.

Evaluation metrics. The performance of method $\mathcal{A}$ on a dataset of size n is measured using the following *progressive validation loss* (Blum et al., 1999):

$$PV_{\mathcal{A}} = \frac{1}{n} \sum_{t=1}^{n} \mathbb{1}\{a_t \neq y_t\},$$

where a_t the action chosen by the algorithm on the t-th example, and y_t the true class label. This metric allows us to capture the explore-exploit trade-off, while providing a measure of generalization that is independent of the choice of loss encodings, and comparable with online supervised learning. We also consider a *normalized loss* variant:

$$NPV_{\mathcal{A}} = \frac{PV_{\mathcal{A}} - PV_{\text{OAA}}}{PV_{\text{OAA}}},$$

where OAA denotes an online (supervised) one against all classifier. This helps highlight the difficulty of exploration for some datasets in our plots.

Win/loss statistics. In order to compare two methods on a given dataset, we consider a notion of statistically significant win or loss. We use the following definition of significance based on a Z-test: if p_a and p_b denote the PV loss of a and b on a given dataset of size n, then a wins over b if

$$1 - \Phi\left(\frac{p_a - p_b}{\sqrt{\frac{p_a(1-p_a)}{n} + \frac{p_b(1-p_b)}{n}}}\right) < 0.05,$$

↓ vs →	G	C-nu	B	B-g	ϵG	C-u	A
G	-	-10	73	73	114	201	84
C-nu	10	-	83	79	150	249	107
B	-73	-83	-	-10	58	162	38
B-g	-73	-79	10	-	74	168	51
ϵG	-114	-150	-58	-74	-	105	-27
C-u	-201	-249	-162	-168	-105	-	-129
A	-84	-107	-38	-51	27	129	-

(a) baseline, reduction, and other hyperparameters optimized

↓ vs →	G-iwr	G-dr	C-nu	B	B-g	ϵG	C-u	A
G-iwr	-	-44	-10	99	99	95	223	86
G-dr	44	-	31	146	143	138	236	121
C-nu	10	-31	-	91	91	110	261	111
B	-99	-146	-91	-	-3	-7	173	14
B-g	-99	-143	-91	3	-	-1	179	24
ϵG	-95	-138	-110	7	1	-	152	11
C-u	-223	-236	-261	-173	-179	-152	-	-175
A	-86	-121	-111	-14	-24	-11	175	-

(b) baseline, reduction, and other hyperparameters fixed by Table 2

Table 3: Each (row, column) entry shows the statistically significant win-loss difference of row against column. The encoding is fixed to -1/0. In (b), only the learning rate is optimized.

where Φ is the Gauss error function. We have found these metrics to provide more insight into the behavior of different methods, compared to strategies based on aggregation of loss measures across all datasets. Indeed, we often found the relative performance of two methods to vary significantly across datasets, making aggregate metrics less appealing. We define the *significant win-loss difference* of one algorithm against another to be the difference between the number of significant wins and significant losses.

4.2 Reducing Exploration

The success of no exploration. Our experiments suggest that the Greedy approach, which has no explicit exploration, performs very well on many of the datasets we consider. This can be seen in Table 3, where the number of significant wins of greedy approaches against other methods generally exceeds that of significant losses (positive numbers in the rows of G, G-iwr, G-dr), and this win-loss difference can be especially large

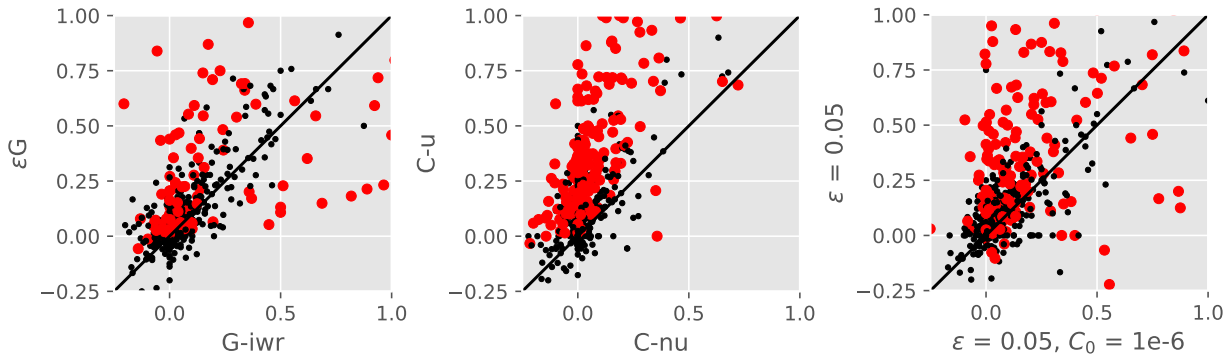


Figure 2: Uniform exploration degrades performance on a large number of datasets. (left) Greedy vs ϵ-greedy; (middle) Cover-nu vs Cover-u; (right) active ϵ-greedy vs ϵ-greedy (DR, -1/0). The plots consider normalized loss, with red points indicating significant wins.

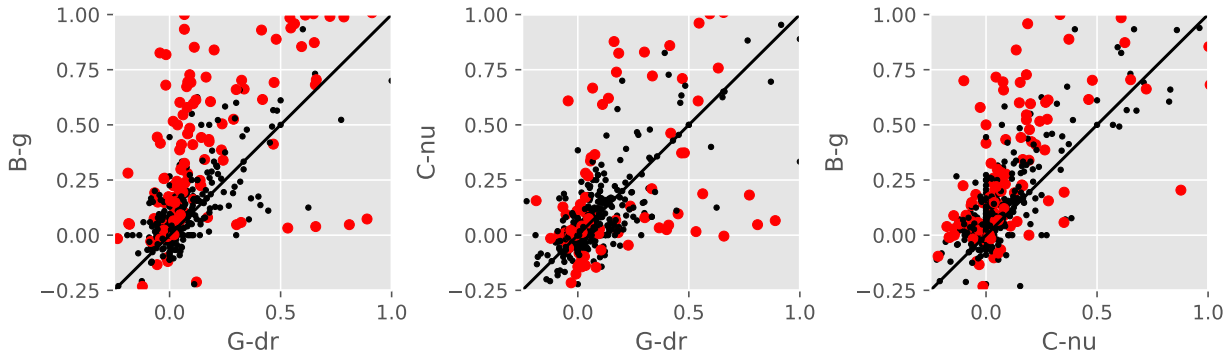


Figure 3: Comparisons among three successful methods: Greedy-DR, Cover-nu, Bag-greedy. Hyperparameters fixed as in Table 2.

against methods with uniform exploration such as ϵ-greedy and Cover-u. More generally, if we consider the Greedy-dr method with -1/0+b, we find that on over 400 datasets out of 524, no other method highlighted in Table 2 wins significantly against it. Figure 2 shows more evidence that uniform exploration may not be needed on many datasets by showing a scatterplot comparing Greedy and ϵ-greedy. A possible reason for this success is the diversity that is inherently present in the distribution of contexts across actions, which has been shown to yield no-regret guarantees under various assumptions (Bastani et al., 2017; Kannan et al., 2018). The noise induced by the dynamics of online learning and the shared baseline may also be a source of more exploration. However, many datasets remain where the natural exploration achieved by Greedy is not sufficient, prompting our work on making exploration algorithms more competitive.

Bag and Cover improvements. The bagging approach is the top-performing method on a few of the more difficult and larger datasets in our collection (including the KDD cup 1999 dataset), but is often outperformed by Greedy on many "easy" datasets that require little to no exploration. The Bag-greedy optimization can mitigate this drawback and often improves on Bag (see Table 3), but the variance induced by online bootstrap sampling is likely too large to compete with Greedy on some easy datasets. In contrast, the simple modification Cover-nu of Cover, which removes uniform exploration as described in Section 3.3, manages to outperform its uniform counterpart on almost every dataset (see Figure 2), while being competitive with other algorithms, as shown in Table 3 for -1/0 encodings (see Table 8 for 0/1 and Table 9 for other choices, in Appendix B). Overall, Figure 3 shows detailed scatterplots for comparing these methods which achieve reduced exploration. While Bag-greedy is often outperformed by both Cover-nu and Greedy, Cover-nu is a good candidate when Greedy is too risky of a choice.

Active ϵ-greedy. Because of its simplicity, ϵ-greedy is often the method of choice in practice, yet the uniform randomization over all actions can be inefficient. Figure 2(right) shows that the active ϵ-greedy strategy introduced in Section 3.4 can be very effective at reducing the level of exploration compared to the corresponding ϵ-greedy method, at a relatively cheap additional cost (see also Figure 5 in the Appendix for more plots with different values of C_0).

4.3 Reductions, encodings, and *baseline*

IWR. The IWR reduction introduced in Section 2.3 has desirable properties, such as a computational cost that is independent of the total number of actions, online requiring updates for the chosen action. In addition to Greedy, we found IWR to work very well for bagging and ϵ-greedy approaches, as shown in Table 4. This may be attributed to the difficulty of the CSC problem compared to regression, as well as importance weight aware online updates, which can be helpful for small ϵ. For bagging, IWR is never significantly outperformed by IPS or DR (see Table 6 in Appendix B), while the domination is not as clear for ϵ-greedy, but together with

↓ vs →	ips	dr	iwr
ips	-	-58	-219
dr	58	-	-194
iwr	219	194	-

↓ vs →	ips	dr	iwr
ips	-	62	-114
dr	-62	-	-135
iwr	114	135	-

Table 4: Impact of reductions for Bag (left) and ϵ-greedy (right), with hyperparameters optimized and encoding fixed by Table 2.

↓ vs →	01	01b	-10	-10b
01	-	-13	-23	-58
01b	13	-	5	-25
-10	23	-5	-	-29
-10b	58	25	29	-

↓ vs →	01	01b	-10	-10b
01	-	61	-30	32
01b	-61	-	-79	-33
-10	30	79	-	50
-10b	-32	33	-50	-

(a) all datasets

↓ vs →	01	01b	-10	-10b
01	-	-7	3	-11
01b	7	-	4	-6
-10	-3	-4	-	-8
-10b	11	6	8	-

↓ vs →	01	01b	-10	-10b
01	-	29	4	30
01b	-29	-	-26	5
-10	-4	26	-	25
-10b	-30	-5	-25	-

(b) datasets with more than 10000 examples (61 datasets)

Table 5: Impact of encoding/baseline for Greedy (left) and Cover-nu (right), with DR, and the remaining hyperparameters optimized.

its computational benefits, our results suggest that IWR is often a compelling alternative to CSC reductions based on IPS or DR. In particular, when the number of actions is prohibitively large, Bag with IWR may be a good default choice of exploration algorithm. While Cover-nu and active ϵ-greedy do not directly support IWR, making them work together well would be a promising future direction.

Encodings and baseline. We found the best choices of encodings and baseline to often be coupled, especially for greedy approaches, hence Table 5 looks at all combinations.

Table 5(a) shows that the -1/0 encoding often outperforms 0/1 for both Greedy and Cover-nu approaches with DR estimates. We now give one possible explanation. As discussed in Section 2.4, the -1/0 encoding yields low variance loss estimates when the 0/1 loss is often 1. Since the learner may often be wrong in early iterations, a loss of 1 is a good initial guess, and the loss estimates may then adapt to observed data as the learning progresses. With enough data, however, the learner should reach better accuracies and observe losses closer to 0, in which case the 0/1 encoding should lead to lower variance estimates. Table 5(b) shows that on large enough datasets 0/1 can indeed be preferable for both Greedy (with no baseline) and Cover-nu.

Table 5 also suggests that baseline along with the -1/0 encoding typically yields the best performance for greedy, which may be attributed to good initial behavior achieved by the -1/0 encoding, together with the fast adaptation to constant loss estimates that baseline helps achieve through a separate online update. In contrast, the benefits of baseline for Cover-nu are less clear, and the -1/0 encoding without baseline is a good default choice.

Robustness to loss range. In an online learning setting, *baseline* can also help to quickly reach an unknown target range of loss estimates. This is demonstrated in Figure 4, where the addition of baseline is shown to help various methods with 9/10 encodings on a large number of datasets.

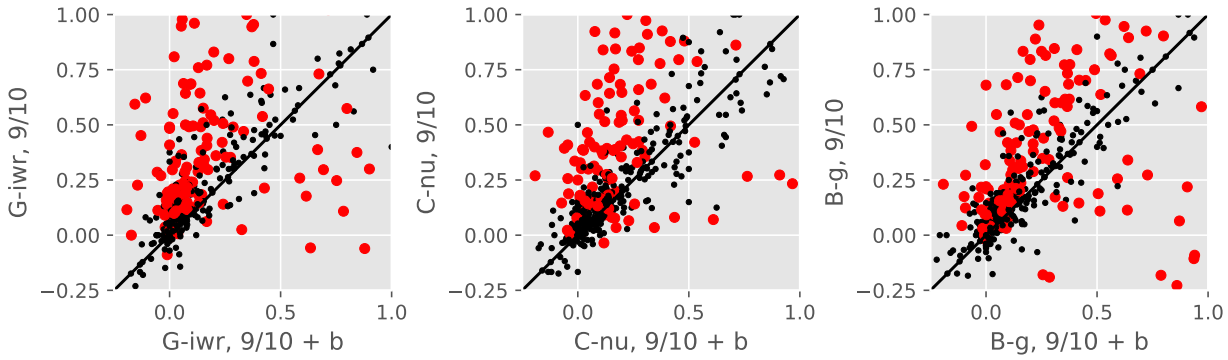


Figure 4: Baseline improves robustness to the range of losses.

5 Discussion and Concluding Remarks

In this paper, we presented an evaluation of practical contextual bandit algorithms on a large collection of multiclass classification datasets with simulated bandit feedback, and proposed optimizations for improved empirical performance. For practitioners, our study gives insight into the relative behavior of different algorithms and learning mechanisms, and the gains that can be obtained with practical modifications, while stressing the importance of loss estimation and other design choices such as how to encode observed feedback. A highlight of our work is that reducing the amount of exploration is often crucial to good practical performance, and that even relying on the exploration induced by the diversity context can often suffice, prompting algorithm designers and theoreticians to develop improved methods which may better exploit this natural exploration phenomenon, better integrate the loss estimation task into the algorithm design, and to aim for improved data-dependent guarantees.

While our evaluation captures key aspects of interactive machine learning settings, some caveats remain when compared to real-world settings. In particular, our setup ignores the non-stationary nature of many real-world settings, the need for offline policy evaluation, and our study only considers linearly parameterized policies, a relatively small number of actions, and simple cost structures. Considering a more general setup would likely require more exploration than in our setting, and an evaluation on real-world interactive systems is an important next step.

References

A. Agarwal, M. Dudík, S. Kale, J. Langford, and R. E. Schapire. Contextual bandit learning with predictable rewards. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2012.

A. Agarwal, D. Hsu, S. Kale, J. Langford, L. Li, and R. E. Schapire. Taming the monster: A fast and simple algorithm for contextual bandits. *arXiv preprint arXiv:1402.0555*, 2014.

A. Agarwal, S. Bird, M. Cozowicz, L. Hoang, J. Langford, S. Lee, J. Li, D. Melamed, G. Oshri, O. Ribas, et al. A multiworld testing decision service. *arXiv preprint arXiv:1606.03966*, 2016.

A. Agarwal, A. Krishnamurthy, J. Langford, H. Luo, et al. Open problem: First-order regret bounds for contextual bandits. In *Conference on Learning Theory (COLT)*, 2017.

S. Agrawal and N. Goyal. Thompson sampling for contextual bandits with linear payoffs. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2013.

H. Bastani, M. Bayati, and K. Khosravi. Exploiting the natural exploration in contextual bandits. *arXiv preprint arXiv:1704.09011*, 2017.

A. Blum, A. Kalai, and J. Langford. Beating the hold-out: Bounds for k-fold and progressive cross-validation. In *Conference on Learning Theory (COLT)*, 1999.

O. Chapelle and L. Li. An empirical evaluation of thompson sampling. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research (JMLR)*, 12(Jul):2121–2159, 2011.

M. Dudik, D. Hsu, S. Kale, N. Karampatziakis, J. Langford, L. Reyzin, and T. Zhang. Efficient optimal learning for contextual bandits. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2011a.

M. Dudik, J. Langford, and L. Li. Doubly robust policy evaluation and learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2011b.

D. Eckles and M. Kaptein. Thompson sampling with the online bootstrap. *arXiv preprint arXiv:1410.4009*, 2014.

S. Hanneke. Theory of disagreement-based active learning. *Foundations and Trends in Machine Learning*, 7 (2-3), 2014.

X. He, J. Pan, O. Jin, T. Xu, B. Liu, T. Xu, Y. Shi, A. Atallah, R. Herbrich, S. Bowers, et al. Practical lessons from predicting clicks on ads at facebook. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*, 2014.

D. J. Hsu. *Algorithms for active learning*. PhD thesis, UC San Diego, 2010.

T.-K. Huang, A. Agarwal, D. J. Hsu, J. Langford, and R. E. Schapire. Efficient and parsimonious agnostic active learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

S. M. Kakade and A. Tewari. On the generalization ability of online strongly convex programming algorithms. In *Advances in Neural Information Processing Systems (NIPS)*, 2009.

S. Kannan, J. Morgenstern, A. Roth, B. Waggoner, and Z. S. Wu. A smoothed analysis of the greedy algorithm for the linear contextual bandit problem. *arXiv preprint arXiv:1801.03423*, 2018.

N. Karampatziakis and J. Langford. Online importance weight aware updates. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2011.

A. Krishnamurthy, A. Agarwal, T.-K. Huang, H. Daume III, and J. Langford. Active learning for cost-sensitive classification. *arXiv preprint arXiv:1703.01014*, 2017.

J. Langford and T. Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in Neural Information Processing Systems (NIPS)*, 2008.

P. Massart, É. Nédélec, et al. Risk bounds for statistical learning. *The Annals of Statistics*, 34(5), 2006.

H. B. McMahan, G. Holt, D. Sculley, M. Young, D. Ebner, J. Grady, L. Nie, T. Phillips, E. Davydov, D. Golovin, et al. Ad click prediction: a view from the trenches. In *Proceedings of the 19th ACM international conference on Knowledge discovery and data mining (KDD)*, 2013.

I. Osband and B. Van Roy. Bootstrapped thompson sampling and deep exploration. *arXiv preprint arXiv:1507.00300*, 2015.

N. C. Oza and S. Russell. Online bagging and boosting. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2001.

Z. Qin, V. Petricek, N. Karampatziakis, L. Li, and J. Langford. Efficient online bootstrapping for large scale learning. In *Workshop on Parallel and Large-scale Machine Learning (BigLearning@NIPS)*, 2013.

S. Ross, P. Mineiro, and J. Langford. Normalized online learning. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2013.

D. Russo, B. Van Roy, A. Kazerouni, and I. Osband. A tutorial on thompson sampling. *arXiv preprint arXiv:1707.02038*, 2017.

W. R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4), 1933.

A Active ϵ-greedy: Practical Implementation and Analysis

A.1 Implementation Details

In this section, we present a practical way to implement the disagreement tests in the active ϵ-greedy method, in the context of online cost-sensitive classification oracles based on regression, as in Vowpal Wabbit. More precisely, we describe the `loss_diff` method of Algorithm 5 used in our experiments.

Let $\hat{L}_{t-1}(\pi)$ denote the empirical loss of policy π on the (biased) sample of cost-sensitive examples collected up to time $t-1$ (see Section A.2 for details). After observing a context x_t, we want to estimate

$$\texttt{loss\_diff}(\pi_t, x_t, \bar{a}) \approx \hat{L}_{t-1}(\pi_{t,\bar{a}}) - \hat{L}_{t-1}(\pi_t),$$

for any action $\bar{a}$, where

$$\pi_t = \arg\min_{\pi} \hat{L}_{t-1}(\pi)$$
$$\pi_{t,\bar{a}} = \arg \min_{\pi:\pi(x_t)=\bar{a}} \hat{L}_{t-1}(\pi).$$

In our online setup, we take π_t to be the current online policy (as in Algorithm 5), and we estimate the loss difference by looking at how many online CSC examples of the form $\bar{c} := (\mathbb{1}\{a \neq \bar{a}\})_{a=1..K}$ are needed (or the importance weight on such an example) in order to switch prediction from $\pi_t(x_t)$ to $\bar{a}$. If we denote this importance weight by $\tau_{\bar{a}}$, then we can estimate $\hat{L}_{t-1}(\pi_{t,\bar{a}}) - \hat{L}_{t-1}(\pi_t) \approx \tau_{\bar{a}}/t$.

Computing $\tau_{\bar{a}}$ for IPS/DR. In the case of IPS/DR, we use an online CSC oracle, which is based on K regressors $f(x,a)$ in VW, each predicting the cost for an action a. Let f_t be the current regressors for policy π_t, $y_t(a) := f_t(x_t, a)$, and denote by $s_t(a)$ the *sensitivity* of regressor $f_t(\cdot, a)$ on example $(x_t, \bar{c}(a))$. This sensitivity is essentially defined to be the derivative with respect to an importance weight w of the prediction $y'(a)$ obtained from the regressor after an online update $(x_t, \bar{c}(a))$ with importance weight w. A similar quantity has been used, *e.g.*, in Huang et al. (2015); Karampatziakis and Langford (2011); Krishnamurthy et al. (2017). Then, the predictions on actions $\bar{a}$ and a cross when the importance weight w satisfies $y_t(\bar{a}) - s_t(\bar{a})w = y_t(a) + s_t(a)w$. Thus, the importance weight required for action $\bar{a}$ to be preferred (*i.e.*, smaller predicted loss) to action a is given by:

$$w_{\bar{a}}^a = \frac{y_t(\bar{a}) - y_t(a)}{s_t(\bar{a}) + s_t(a)}.$$

Action $\bar{a}$ will thus be preferred to all other actions when using an importance weight $\tau_{\bar{a}} = \max_a w_{\bar{a}}^a$.

Computing $\tau_{\bar{a}}$ for IWR. Although Algorithm 5 and the theoretical analysis require CSC in order to assign a loss of 1 to unexplored actions, and hence does not directly support IWR, we can consider an approximation which leverages the benefits of IWR by performing standard IWR updates as in ϵ-greedy, while exploring only on actions that pass a similar disagreement test. In this case, we estimate $\tau_{\bar{a}}$ as the importance weight on an online regression example $(x_t, 0)$ for the regressor $f_t(\cdot, \bar{a})$, needed to switch prediction to $\bar{a}$. If $s_t(\bar{a})$ is the sensitivity for such an example, we have $\tau_{\bar{a}} = (y_t(\bar{a}) - y_t^*)/s_t(\bar{a})$, where $y_t^* = \min_a y_t(a)$.

A.2 Theoretical Analysis

This section presents a theoretical analysis of the active ϵ-greedy method introduced in Section 3.4. We begin by presenting the analyzed version of the algorithm together with definitions in Section A.2.1. Section A.2.2 then studies the correctness of the method, showing that with high probability, the actions chosen by the optimal policy are always explored, and that policies considered by the algorithm are always as good as those obtained under standard ϵ-greedy exploration. This section also introduces a Massart-type low-noise condition similar to the one considered in (Krishnamurthy et al., 2017) for cost-sensitive classification. Finally,

Section A.2.3 provides a regret analysis of the algorithm, both in the worst case and under disagreement conditions together with the Massart noise condition. In particular, a formal version of Theorem 1 is given by Theorem 8, and a more extreme but informative situation is considered in Proposition 9, where our algorithm can achieve constant regret.

A.2.1 Algorithm and definitions

We consider a version of the active ϵ-greedy strategy that is more suitable for theoretical analysis, given in Algorithm 6. This method considers exact CSC oracles, as well as a CSC oracle with one constraint on the policy ($\pi(x_t) = a$ in Eq.(9)). The threshold Δ_t is defined later in Section A.2.2. Computing it would require some knowledge about the size of the policy class, which we avoid by introducing a parameter C_0 in the practical variant. The disagreement strategy is based on the Oracular CAL active learning method of (Hsu, 2010), which tests for disagreement using empirical error differences, and considers biased samples when no label is queried. Here, similar tests are used to decide which actions should be explored, in the different context of cost-sensitive classification, and the unexplored actions are assigned a loss of 1, making the empirical sample biased ($\hat{Z}_T$ in Algorithm 6).

Definitions. Define $Z_T = \{(x_t, \ell_t)\}_{t=1..T} \subset \mathcal{X} \times \mathbb{R}^K$, $\tilde{Z}_T = \{(x_t, \tilde{\ell}_t)\}_{t=1..T}$ (biased sample) and $\hat{Z}_T = \{(x_t, \hat{\ell}_t)\}_{t=1..T}$ (IPS estimate of biased sample), where $\ell_t \in [0,1]^K$ is the (unobserved) loss vector at time t and

$$\tilde{\ell}_t(a) = \begin{cases} \ell_t(a), & \text{if } a \in A_t \\ 1, & \text{o/w} \end{cases} \tag{7}$$

$$\hat{\ell}_t(a) = \begin{cases} \frac{\mathbb{1}\{a=a_t\}}{p_t(a_t)} \ell_t(a_t), & \text{if } a \in A_t \\ 1, & \text{o/w.} \end{cases} \tag{8}$$

For any set $Z \subset \mathcal{X} \times \mathbb{R}^K$ defined as above, we denote, for $\pi \in \Pi$,

$$L(\pi, Z) = \frac{1}{|Z|} \sum_{(x,c) \in Z} c(\pi(x)).$$

We then define the empirical losses $L_T(\pi) := L(\pi, Z_T)$, $\hat{L}_T(\pi) := L(\pi, \hat{Z}_T)$ and $\tilde{L}_T(\pi) := L(\pi, \tilde{Z}_T)$. Let $L(\pi) := \mathbb{E}_{(x,\ell)\sim D}[\ell(\pi(x))]$ be the expected loss of policy π, and $\pi^* := \arg\min_{\pi \in \Pi} L(\pi)$. We also define $\rho(\pi, \pi') := P_x(\pi(x) \neq \pi'(x))$, the expected disagreement between policies π and π', where P_x denotes the marginal distribution of D on contexts.

A.2.2 Correctness

We begin by stating a lemma that controls deviations of empirical loss differences, which relies on Freedman's inequality for martingales (see, *e.g.*, Kakade and Tewari, 2009, Lemma 3).

Lemma 2 (Deviation bounds)**.** *With probability $1-\delta$, the following event holds: for all $\pi \in \Pi$, for all $T \geq 1$,*

$$|(\hat{L}_T(\pi) - \hat{L}_T(\pi^*)) - (\tilde{L}_T(\pi) - \tilde{L}_T(\pi^*))| \leq \sqrt{\frac{2K\rho(\pi,\pi^*)e_T}{\epsilon}} + \left(\frac{K}{\epsilon} + 1\right) e_T \tag{11}$$

$$|(L_T(\pi) - L_T(\pi^*)) - (L(\pi) - L(\pi^*))| \leq \sqrt{\rho(\pi,\pi^*)e_T} + 2e_T, \tag{12}$$

where $e_T = \log(2|\Pi|/\delta_T)/T$ and $\delta_T = \delta/(T^2+T)$. We denote this event by $\mathcal{E}$ in what follows.

Proof. We prove the result using Freedman's inequality (see, *e.g.*, Kakade and Tewari, 2009, Lemma 3), which controls deviations of a sum using the conditional variance of each term in the sum and an almost sure bound on their magnitude, along with a union bound.

Algorithm 6 ϵ-greedy with disagreement tests

Input: exploration probability ϵ.
Initialize: $\hat{Z}_0 := \emptyset$.
for $t = 1, \ldots$ **do**
Observe context x_t. Let

$$\pi_t := \arg\min_{\pi} L(\pi, \hat{Z}_{t-1})$$
$$\pi_{t,a} := \arg\min_{\pi:\pi(x_t)=a} L(\pi, \hat{Z}_{t-1}) \tag{9}$$
$$A_t := \{a : L(\pi_{t,a}, \hat{Z}_{t-1}) - L(\pi_t, \hat{Z}_{t-1}) \leq \Delta_t\} \tag{10}$$

Let

$$p_t(a) = \begin{cases} 1 - (|A_t| - 1)\epsilon/K, & \text{if } a = \pi_t(x_t) \\ \epsilon/K, & \text{if } a \in A_t \setminus \{\pi_t(x_t)\} \\ 0, & \text{otherwise.} \end{cases}$$

Play action $a_t \sim p_t$, observe $\ell_t(a_t)$ and set $\hat{Z}_t = \hat{Z}_{t-1} \cup \{(x_t, \hat{\ell}_t)\}$, where $\hat{\ell}_t$ is defined in (8).
end for

For (11), let $(\hat{L}_T(\pi) - \hat{L}_T(\pi^*)) - (\tilde{L}_T(\pi) - \tilde{L}_T(\pi^*)) = \frac{1}{T}\sum_{t=1}^T R_t$, with

$$R_t = \hat{\ell}_t(\pi(x_t)) - \hat{\ell}_t(\pi^*(x_t)) - (\tilde{\ell}_t(\pi(x_t)) - \tilde{\ell}_t(\pi^*(x_t))).$$

We define the σ-fields $\mathcal{F}_t := \sigma(\{x_i, \ell_i, a_i\}_{i=1}^t)$. Note that R_t is $\mathcal{F}_t$-measurable and

$$\mathbb{E}[\hat{\ell}_t(\pi(x_t)) - \hat{\ell}_t(\pi^*(x_t))|x_t, \ell_t] = \tilde{\ell}_t(\pi(x_t)) - \tilde{\ell}_t(\pi^*(x_t)),$$

so that $\mathbb{E}[R_t|\mathcal{F}_{t-1}] = \mathbb{E}[\mathbb{E}[R_t|x_t, \ell_t]|\mathcal{F}_{t-1}] = 0$. Thus, $(R_t)_{t\geq 1}$ is a martingale difference sequence adapted to the filtration $(\mathcal{F}_t)_{t\geq 1}$. We have

$$|R_t| \leq |\hat{\ell}_t(\pi(x_t)) - \hat{\ell}_t(\pi^*(x_t))| + |\tilde{\ell}_t(\pi(x_t)) - \tilde{\ell}_t(\pi^*(x_t))| \leq \frac{K}{\epsilon} + 1.$$

Note that $\mathbb{E}[\hat{\ell}_t(\pi(x_t)) - \hat{\ell}_t(\pi^*(x_t))|x_t, \ell_t] = \tilde{\ell}_t(\pi(x_t)) - \tilde{\ell}_t(\pi^*(x_t))$, so that

$$\begin{aligned}
\mathbb{E}[R_t^2|\mathcal{F}_{t-1}] &= \mathbb{E}[\mathbb{E}[R_t^2|x_t, \ell_t, A_t]|\mathcal{F}_{t-1}] \\
&\leq \mathbb{E}[\mathbb{E}[(\hat{\ell}_t(\pi(x_t)) - \hat{\ell}_t(\pi^*(x_t)))^2|x_t, \ell_t, A_t]|\mathcal{F}_{t-1}] \\
&\leq \mathbb{E}\left[\mathbb{E}\left[\frac{(\mathbb{1}\{\pi(x_t) = a_t\} - \mathbb{1}\{\pi^*(x_t) = a_t\})^2}{p_t(a_t)^2}|x_t, \ell_t, A_t\right]|\mathcal{F}_{t-1}\right] \\
&\leq \mathbb{E}\left[\mathbb{E}\left[\frac{\mathbb{1}\{\pi(x_t) \neq \pi^*(x_t)\}(\mathbb{1}\{\pi(x_t) = a_t\} + \mathbb{1}\{\pi^*(x_t) = a_t\})}{p_t(a_t)^2}|x_t, \ell_t, A_t\right]|\mathcal{F}_{t-1}\right] \\
&= \mathbb{E}\left[\frac{2K\ \mathbb{1}\{\pi(x_t) \neq \pi^*(x_t)\}}{\epsilon}|\mathcal{F}_{t-1}\right] = \frac{2K}{\epsilon}\rho(\pi, \pi^*).
\end{aligned}$$

Freedman's inequality then states that (11) holds with probability $1 - \delta_T/2|\Pi|$.

For (12), we consider a similar setup with

$$R_t = \ell_t(\pi(x_t)) - \ell_t(\pi^*(x_t)) - (L(\pi) - L(\pi^*)).$$

We have $\mathbb{E}[R_t|\mathcal{F}_{t-1}] = 0$, $|R_t| \leq 2$ and $\mathbb{E}[R_t^2|\mathcal{F}_{t-1}] \leq \rho(\pi, \pi^*)$, which yields that (12) holds with probability $1 - \delta_T/2|\Pi|$ using Freedman's inequality. A union bound on $\pi \in \Pi$ and $T \geq 1$ gives the desired result. $\square$

Threshold. We define the threshold Δ_T used in (10) in Algorithm 6 as:

$$\Delta_T := \left(\sqrt{\frac{2K}{\epsilon}} + 1\right)\sqrt{e_{T-1}} + \left(\frac{K}{\epsilon} + 3\right)e_{T-1}. \tag{13}$$

We also define the following more precise deviation quantity for a given policy, which follows directly from the deviation bounds in Lemma 2

$$\Delta_T^*(\pi) := \left(\sqrt{\frac{2K}{\epsilon}} + 1\right)\sqrt{\rho(\pi,\pi^*)e_{T-1}} + \left(\frac{K}{\epsilon} + 3\right)e_{T-1}. \tag{14}$$

Note that we have $\Delta_T^*(\pi) \le \Delta_T$ for any policy π.

The next lemma shows that the bias introduced in the empirical sample by assigning a loss of 1 to unexplored actions is favorable, in the sense that it will not hurt us in identifying π^*.

Lemma 3 (Favorable bias)**.** *Assume $\pi^*(x_t) \in A_t$ for all $t \le T$. We have*

$$\tilde{L}_T(\pi) - \tilde{L}_T(\pi^*) \ge L_T(\pi) - L_T(\pi^*). \tag{15}$$

Proof. For any $t \le T$, we have $\tilde{\ell}_t(a) \ge \ell_t(a)$, so that $\tilde{L}_T(\pi) \ge L_T(\pi)$. Separately, we have $\tilde{\ell}_t(\pi^*(x_t)) = \ell_t(\pi^*(x_t))$ for all $t \le T$ using the definition of $\tilde{\ell}_t$ and the assumption $\pi^*(x_t) \in A_t$, hence $\tilde{L}_T(\pi^*) \ge L_T(\pi^*)$. □

We now show that with high probability, the optimal action is always explored by the algorithm.

Lemma 4. *Assume that event $\mathcal{E}$ holds. The actions given by the optimal policy are always explored for all $t \ge 1$, i.e., $\pi^*(x_t) \in A_t$ for all $t \ge 1$.*

Proof. We show by induction on $T \ge 1$ that $\pi^*(x_t) \in A_t$ for all $t = 1, \dots, T$. For the base case, we have $A_1 = [K]$ since $\hat{Z}_0 = \emptyset$ and hence empirical errors are always equal to 0, so that $\pi^*(x_1) \in A_1$. Let us now assume as the inductive hypothesis that $\pi^*(x_t) \in A_t$ for all $t \le T-1$.

From deviation bounds, we have

$$\hat{L}_{T-1}(\pi_T) - \hat{L}_{T-1}(\pi^*) \ge \tilde{L}_{T-1}(\pi_T) - \tilde{L}_{T-1}(\pi^*) - \left(\sqrt{\frac{2K\rho(\pi,\pi^*)e_{T-1}}{\epsilon}} + (K/\epsilon + 1)e_{T-1}\right)$$

$$L_{T-1}(\pi_T) - L_{T-1}(\pi^*) \ge L(\pi_T) - L(\pi^*) - \left(\sqrt{\rho(\pi,\pi^*)e_{T-1}} + 2e_{T-1}\right).$$

Using Lemma 3 together with the inductive hypothesis, the above inequalities yield

$$\hat{L}_{T-1}(\pi_T) - \hat{L}_{T-1}(\pi^*) \ge L(\pi_T) - L(\pi^*) - \Delta_T^*(\pi_T).$$

Now consider an action $a \notin A_t$. Using the definition (10) of A_t, we have

$$\begin{aligned}\hat{L}_{T-1}(\pi_{T,a}) - \hat{L}_{T-1}(\pi^*) &= \hat{L}_{T-1}(\pi_{T,a}) - \hat{L}_{T-1}(\pi_T) + \hat{L}_{T-1}(\pi_T) - \hat{L}_{T-1}(\pi^*) \\ &> \Delta_T - \Delta_T^*(\pi_T) = 0,\end{aligned}$$

which implies $\pi^*(x_T) \ne a$, since $\hat{L}_{T-1}(\pi_{T,a})$ is the minimum of $\hat{L}_{T-1}$ over policies satisfying $\pi(x_T) = a$. This yields $\pi^*(x_T) \in A_T$, which concludes the proof. □

With the previous results, we can now prove that with high probability, discarding some of the actions from the exploration process does not hurt us in identifying good policies. In particular, π_{T+1} is about as good as it would have been with uniform ϵ-exploration all along.

Theorem 5. *Under the event $\mathcal{E}$, which holds with probability $1-\delta$,*

$$L(\pi_{T+1}) - L(\pi^*) \le \Delta_{T+1}^*(\pi_{T+1}).$$

In particular, $L(\pi_{T+1}) - L(\pi^*) \le \Delta_{T+1}$.

Proof. Assume event $\mathcal{E}$ holds. Using (11-12) combined with Lemma 3 (which holds by Lemma 4), we have

$$L(\pi_{T+1}) - L(\pi^*) \leq \hat{L}_T(\pi_{T+1}) - \hat{L}_T(\pi^*) + \Delta^*_{T+1}(\pi_{T+1}) \leq \Delta^*_{T+1}(\pi_{T+1}).$$

□

Massart noise condition. We introduce a low-noise condition that will help us obtain improved regret guarantees. Similar conditions have been frequently used in supervised learning (Massart et al., 2006) and active learning (Hsu, 2010; Huang et al., 2015; Krishnamurthy et al., 2017) for obtaining better data-dependent guarantees. We consider the following Massart noise condition with parameter $\tau > 0$:

$$\rho(\pi, \pi^*) \leq \frac{1}{\tau}(L(\pi) - L(\pi^*)). \tag{M}$$

This condition holds when $\mathbb{E}[\min_{a \neq \pi^*(x)} \ell(a) - \ell(\pi^*(x))|x] \geq \tau$, P_x-almost surely, which is similar to the Massart condition considered in (Krishnamurthy et al., 2017) in the context of active learning for cost-sensitive classification. Indeed, we have

$$\begin{aligned}
L(\pi) - L(\pi^*) &= \mathbb{E}[\mathbb{1}\{\pi(x) \neq \pi^*(x)\}(\ell(\pi(x)) - \ell(\pi^*(x)) + \mathbb{1}\{\pi(x) = \pi^*(x)\}(\ell(\pi^*(x)) - \ell(\pi^*(x)))] \\
&\geq \mathbb{E}\left[\mathbb{1}\{\pi(x) \neq \pi^*(x)\}\left(\min_{a \neq \pi^*(x)} \ell(a) - \ell(\pi^*(x))\right)\right] \\
&= \mathbb{E}[\mathbb{1}\{\pi(x) \neq \pi^*(x)\}\,\mathbb{E}[\min_{a \neq \pi^*(x)} \ell(a) - \ell(\pi^*(x))|x]] \\
&\geq \mathbb{E}[\mathbb{1}\{\pi(x) \neq \pi^*(x)\}\tau] = \tau\rho(\pi, \pi^*),
\end{aligned}$$

which is precisely (M). The condition allows us to obtain a fast rate for the policies considered by our algorithm, as we now show.

Theorem 6. *Assume the Massart condition* (M) *holds with parameter τ. Under the event $\mathcal{E}$, which holds w.p. $1 - \delta$,*

$$L(\pi_{T+1}) - L(\pi^*) \leq C\frac{K}{\tau\epsilon}e_T,$$

for some numeric constant C.

Proof. Using Theorem 5 and the Massart condition, we have

$$\begin{aligned}
L(\pi_{T+1}) - L(\pi^*) \leq \Delta^*_{T+1}(\pi_{T+1}) &= \left(\sqrt{\frac{2K}{\epsilon}} + 1\right)\sqrt{\rho(\pi_{T+1}, \pi^*)e_T} + \left(\frac{K}{\epsilon} + 3\right)e_T \\
&\leq \left(\sqrt{\frac{2K}{\epsilon}} + 1\right)\sqrt{(L(\pi_{T+1}) - L(\pi^*))e_T/\tau} + \left(\frac{K}{\epsilon} + 3\right)e_T \\
&\leq \sqrt{\frac{8Ke_T}{\tau\epsilon}(L(\pi_{T+1}) - L(\pi^*))} + \frac{4Ke_T}{\epsilon}.
\end{aligned}$$

Solving the quadratic inequality in $L(\pi_{T+1}) - L(\pi^*)$ yields the result. □

A.2.3 Regret Analysis

In a worst-case scenario, the following result shows that Algorithm 6 enjoys a similar $O(T^{2/3})$ regret guarantee to the vanilla ϵ-greedy approach (Langford and Zhang, 2008).

Theorem 7. *Conditioned on the event $\mathcal{E}$, which holds with probability $1 - \delta$, the expected regret of the algorithm is*

$$\mathbb{E}[R_T|\mathcal{E}] \leq O\left(\sqrt{\frac{KT\log(T|\Pi|/\delta)}{\epsilon}} + T\epsilon\right).$$

Optimizing over the choice of ϵ yields a regret $O(T^{2/3}(K\log(T|\Pi|/\delta))^{1/3})$.

Proof. We condition on the $1-\delta$ probability event $\mathcal{E}$ that the deviation bounds of Lemma 2 hold. We have

$$\begin{aligned}\mathbb{E}[\ell_t(a_t) - \ell_t(\pi^*(x_t))|\mathcal{F}_{t-1}] &= \mathbb{E}[\mathbb{1}\{a_t = \pi_t(x_t)\}(\ell_t(\pi_t(x_t)) - \ell_t(\pi^*(x_t))) + \mathbb{1}\{a_t \neq \pi_t(x_t)\}(\ell_t(a_t) - \ell_t(\pi^*(x_t)))|\mathcal{F}_{t-1}] \\ &\leq \mathbb{E}[\ell_t(\pi_t(x_t)) - \ell_t(\pi^*(x_t))|\mathcal{F}_{t-1}] + \mathbb{E}[\mathbb{E}[1 - p_t(\pi_t(x_t))|x_t]|\mathcal{F}_{t-1}] \\ &\leq L(\pi_t) - L(\pi^*) + \epsilon.\end{aligned}$$

Summing over t and applying Theorem 5 together with $\Delta_t^*(\pi) \leq \Delta_t$, we obtain

$$\begin{aligned}&\mathbb{E}[R_T|\mathcal{E}]\,\mathbb{E}\left[\sum_{t=1}^{T} \ell_t(a_t) - \ell_t(\pi^*(x_t))|\mathcal{E}\right] \\ &\leq 1 + \sum_{t=2}^{T} \mathbb{E}[L(\pi_t) - L(\pi^*) + \epsilon|\mathcal{F}_{t-1}, \mathcal{E}] \\ &\leq 1 + T\epsilon + \sum_{t=2}^{T} \Delta_t.\end{aligned}$$

Using $\sum_{t=2}^{T} \sqrt{e_t} \leq O(\sqrt{T\log(8T^2|\Pi|/\delta)})$ and $\sum_{t=2}^{T} e_t \leq O(\log(8T^2|\Pi|/\delta)\log T)$, we obtain

$$\mathbb{E}[R_T|\mathcal{E}] \leq O\left(1 + \sqrt{\frac{KT\log(T|\Pi|/\delta)}{\epsilon}} + \frac{K\log(T|\Pi|/\delta)}{\epsilon}\log T + T\epsilon\right),$$

which yields the result. □

Disagreement definitions. In order to obtain improvements in regret guarantees over the worst case, we consider notions of disagreement that extend standard definitions from the active learning literature (*e.g.*, Hanneke, 2014; Hsu, 2010; Huang et al., 2015) to the multiclass case. Let $B(\pi^*, r) := \{\pi \in \Pi : \rho(\pi, \pi^*) \leq r\}$ be the ball centered at π^* under the (pseudo)-metric $\rho(\cdot,\cdot)$. We define the disagreement region $DIS(r)$ and disagreement coefficient θ as follows:

$$\begin{aligned}DIS(r) &:= \{x : \exists \pi \in B(\pi^*, r) \quad \pi(x) \neq \pi^*(x)\} \\ \theta &:= \sup_{r>0} \frac{P(x \in DIS(r))}{r}.\end{aligned}$$

The next result shows that under the Massart condition and with a finite disagreement coefficient θ, our algorithm achieves a regret that scales as $O(T^{1/3})$ (up to logarithmic factors), thus improving on worst-case guarantees obtained by optimal algorithms such as (Agarwal et al., 2012, 2014; Dudik et al., 2011a).

Theorem 8. *Assume the Massart condition* (M) *holds with parameter τ. Conditioning on the event $\mathcal{E}$ which holds w.p. $1-\delta$, the algorithm has expected regret*

$$\mathbb{E}[R_T|\mathcal{E}] \leq O\left(\frac{K\log(T|\Pi|/\delta)}{\tau\epsilon}\log T + \frac{\theta}{\tau}\sqrt{\epsilon KT\log(T|\Pi|/\delta)}\right).$$

Optimizing over the choice of ϵ yields a regret

$$\mathbb{E}[R_T|\mathcal{E}] \leq O\left(\frac{1}{\tau}(\theta K\log(T|\Pi|/\delta))^{2/3}(T\log T)^{1/3}\right).$$

Proof. Assume $\mathcal{E}$ holds. Let $t \geq 2$, and assume $a \in A_t \setminus \{\pi^*(x_t)\}$. Define

$$\pi_a = \begin{cases}\pi_t, & \text{if } \pi_t(x_t) = a \\ \pi_{t,a}, & \text{if } \pi_t(x_t) \neq a,\end{cases}$$

so that we have $\pi_a(x_t) = a \neq \pi^*(x_t)$.

- If $\pi_a = \pi_t$, then $L(\pi_a) - L(\pi^*) \leq \Delta_t^*(\pi_a) \leq \Delta_t$ by Theorem 5
- If $\pi_a = \pi_{t,a}$, using deviation bounds, Lemma 4 and 3, we have

$$\begin{aligned} L(\pi_a) - L(\pi^*) &= L(\pi_{t,a}) - L(\pi^*) \\ &\leq \hat{L}_{t-1}(\pi_{t,a}) - \hat{L}_{t-1}(\pi^*) + \Delta_t^*(\pi_{t,a}) \\ &= \underbrace{\hat{L}_{t-1}(\pi_{t,a}) - \hat{L}_{t-1}(\pi_t)}_{\leq \Delta_t} + \underbrace{\hat{L}_{t-1}(\pi_t) - \hat{L}_{t-1}(\pi^*)}_{\leq 0} + \Delta_t^*(\pi_{t,a}) \\ &\leq 2\Delta_t, \end{aligned}$$

  where the last inequality uses $a \in A_t$.

By the Massart assumption, we then have $\rho(\pi_a, \pi^*) \leq 2\Delta_t/\tau$. Hence, we have $x_t \in DIS(2\Delta_t/\tau)$. We have thus shown

$$\mathbb{E}[\mathbb{E}[\mathbb{1}\{a \in A_t \setminus \{\pi^*(x_t)\}\}|x_t]|\mathcal{F}_{t-1}] \leq \mathbb{E}[P(x_t \in DIS(2\Delta_t/\tau))|\mathcal{F}_{t-1}] \leq 2\theta\Delta_t/\tau.$$

We then have

$$\begin{aligned} \mathbb{E}[\ell_t(a_t) - \ell_t(\pi^*(x_t))|\mathcal{F}_{t-1}] &= \mathbb{E}[\mathbb{1}\{a_t = \pi_t(x_t)\}(\ell_t(\pi_t(x_t)) - \ell_t(\pi^*(x_t))) \\ &\quad + \mathbb{1}\{a_t = \pi^*(x_t) \wedge a_t \neq \pi_t(x_t)\}(\ell_t(\pi^*(x_t)) - \ell_t(\pi^*(x_t))) \\ &\quad + \sum_{a=1}^{K} \mathbb{1}\{a_t = a \wedge a \notin \{\pi_t(x_t), \pi^*(x_t)\}\}(\ell_t(a) - \ell_t(\pi^*(x_t)))|\mathcal{F}_{t-1}] \\ &\leq \mathbb{E}\left[\ell_t(\pi_t(x_t)) - \ell_t(\pi^*(x_t)) + \sum_{a=1}^{K} \mathbb{E}[\mathbb{1}\{a_t = a\}\, \mathbb{1}\{a \notin \{\pi_t(x_t), \pi^*(x_t)\}\}|x_t]|\mathcal{F}_{t-1}\right] \\ &= L(\pi_t) - L(\pi^*) + \sum_{a=1}^{K} \mathbb{E}[\mathbb{E}[p_t(a)\, \mathbb{1}\{a \notin \{\pi_t(x_t), \pi^*(x_t)\}\}|x_t]|\mathcal{F}_{t-1}] \\ &\leq L(\pi_t) - L(\pi^*) + \frac{\epsilon}{K} \sum_{a=1}^{K} \mathbb{E}[\mathbb{E}[\mathbb{1}\{a \in A_t \setminus \{\pi^*(x_t)\}\}|x_t]|\mathcal{F}_{t-1}] \\ &\leq C\frac{K}{\tau\epsilon} e_{t-1} + 2\epsilon\theta\Delta_t/\tau, \end{aligned}$$

where we used

$$\begin{aligned} p_t(a)\, \mathbb{1}\{a \notin \{\pi_t(x_t), \pi^*(x_t)\}\} &= \frac{\epsilon}{K}\, \mathbb{1}\{a \in A_t \setminus \{\pi_t(x_t), \pi^*(x_t)\}\} \\ &\leq \frac{\epsilon}{K}\, \mathbb{1}\{a \in A_t \setminus \{\pi^*(x_t)\}\}. \end{aligned}$$

Summing over t and taking total expectations (conditioned on $\mathcal{E}$) yields

$$\mathbb{E}[R_T|\mathcal{E}] \leq O\left(\frac{K\log(T|\Pi|/\delta)}{\tau\epsilon}\log T + \frac{\epsilon\theta}{\tau}\left(\sqrt{\frac{KT\log(T|\Pi|/\delta)}{\epsilon}} + \frac{K\log(T|\Pi|/\delta)}{\epsilon}\log(T)\right)\right),$$

and the result follows. □

Finally, we look at a simpler instructive example, which considers an extreme situation where the expected loss of any suboptimal policy is bounded away from that of the optimal policy. In this case, Algorithm 6 can achieve constant regret when the disagreement coefficient is bounded, as shown by the following result.

Proposition 9. *Assume that $L(\pi) - L(\pi^*) \geq \tau > 0$ for all $\pi \neq \pi^*$, and that $\theta < \infty$. Under the event $\mathcal{E}$, the algorithm achieves constant expected regret. In particular, the algorithm stops incurring regret for $T > T_0 := \max\{t : 2\Delta_t > \tau\}$.*

Proof. By Theorem 5 and our assumption, we have $L(\pi_t) - L(\pi^*) \leq \mathbb{1}\{\Delta_t \geq \tau\}\Delta_t$. Similarly, the assumption implies that $\rho(\pi, \pi^*) \leq \mathbb{1}\{L(\pi) - L(\pi^*) \geq \tau\}$, so that using similar arguments to the proof of Theorem 8, we have

$$\mathbb{E}[\mathbb{E}[\mathbb{1}\{a \in A_t \setminus \{\pi^*(x_t)\}\}|x_t]|\mathcal{F}_{t-1}] \leq \theta\, \mathbb{1}\{2\Delta_t \geq \tau\}.$$

Following the proof of Theorem 8, this implies that when t is such that $2\Delta_t < \tau$, then we have

$$\mathbb{E}[\ell_t(a_t) - \ell_t(\pi^*(x_t))|\mathcal{F}_{t-1}] = 0.$$

Let $T_0 := \max\{t : 2\Delta_t \geq \tau\}$. We thus have

$$\mathbb{E}[R_T|\mathcal{E}] \leq 1 + \sum_{t=2}^{T_0}(\Delta_t + \epsilon).$$

□

B Additional Results from the Evaluation

This sections provides additional experimental results, and more detailed win/loss statistics for tables in the main paper, showing both significant wins and significant losses, rather than just their difference. Table 8 also provides more results for algorithms across different encodings and with various hyperparameters optimized. Figure 5 shows the improvements that the active ϵ-greedy algorithm can achieve compared to ϵ-greedy, under different settings of encodings and reductions.

List of datasets. The datasets we used can be accessed at `https://www.openml.org/d/<id>`, with `id` in the following list:

3, 6, 8, 10, 11, 12, 14, 16, 18, 20, 21, 22, 23, 26, 28, 30, 31, 32, 36, 37, 39, 40, 41, 43, 44, 46, 48, 50, 53, 54, 59, 60, 61, 62, 150, 151, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 180, 181, 182, 183, 184, 187, 189, 197, 209, 223, 227, 273, 275, 276, 277, 278, 279, 285, 287, 292, 293, 294, 298, 300, 307, 310, 312, 313, 329, 333, 334, 335, 336, 337, 338, 339, 343, 346, 351, 354, 357, 375, 377, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 444, 446, 448, 450, 457, 458, 459, 461, 462, 463, 464, 465, 467, 468, 469, 472, 475, 476, 477, 478, 479, 480, 554, 679, 682, 683, 685, 694, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754, 755, 756, 758, 759, 761, 762, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 782, 783, 784, 785, 787, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 799, 800, 801, 803, 804, 805, 806, 807, 808, 811, 812, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 832, 833, 834, 835, 836, 837, 838, 841, 843, 845, 846, 847, 848, 849, 850, 851, 853, 855, 857, 859, 860, 862, 863, 864, 865, 866, 867, 868, 869, 870, 871, 872, 873, 874, 875, 876, 877, 878, 879, 880, 881, 882, 884, 885, 886, 888, 891, 892, 893, 894, 895, 896, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 910, 911, 912, 913, 914, 915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 931, 932, 933, 934, 935, 936, 937, 938, 941, 942, 943, 945, 946, 947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 958, 959, 962, 964, 965, 969, 970, 971, 973, 974, 976, 977, 978, 979, 980, 983, 987, 988, 991, 994, 995, 996, 997, 1004, 1005, 1006, 1009, 1011, 1012, 1013, 1014, 1015, 1016, 1019, 1020, 1021, 1022, 1025, 1026, 1036, 1038, 1040, 1041, 1043, 1044, 1045, 1046, 1048, 1049, 1050, 1054, 1055, 1056, 1059, 1060, 1061, 1062, 1063, 1064, 1065, 1066, 1067, 1068, 1069, 1071, 1073, 1075, 1077, 1078, 1079, 1080, 1081, 1082, 1083, 1084, 1085, 1086, 1087, 1088, 1100, 1104, 1106, 1107, 1110, 1113, 1115, 1116, 1117, 1120, 1121, 1122, 1123, 1124, 1125, 1126, 1127, 1128, 1129, 1130, 1131, 1132, 1133, 1135, 1136, 1137, 1138, 1139, 1140, 1141, 1142, 1143, 1144, 1145, 1146, 1147, 1148, 1149, 1150, 1151, 1152, 1153, 1154, 1155, 1156, 1157, 1158, 1159, 1160, 1161, 1162, 1163, 1164, 1165, 1166, 1169, 1216, 1217, 1218, 1233, 1235, 1236, 1237, 1238, 1241, 1242, 1412, 1413, 1441, 1442, 1443, 1444, 1449, 1451, 1453, 1454, 1455, 1457, 1459, 1460, 1464, 1467, 1470, 1471, 1472, 1473, 1475, 1481, 1482, 1483, 1486, 1487, 1488, 1489, 1496, 1498.

↓ vs →	ips	dr	iwr
ips	-	77 / 135	0 / 219
dr	135 / 77	-	0 / 194
iwr	219 / 0	194 / 0	-

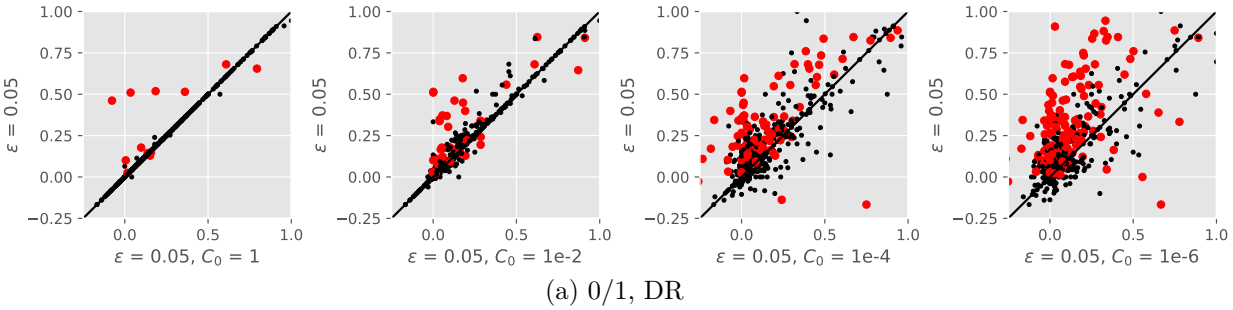
↓ vs →	ips	dr	iwr
ips	-	88 / 26	11 / 125
dr	26 / 88	-	12 / 147
iwr	125 / 11	147 / 12	-

Table 6: Detailed win/loss version of Table 4 with significant wins/losses of row vs column: impact of reductions for Bag (left) and ϵ-greedy (right), with optimized hyperparameters and encoding fixed by Table 2.

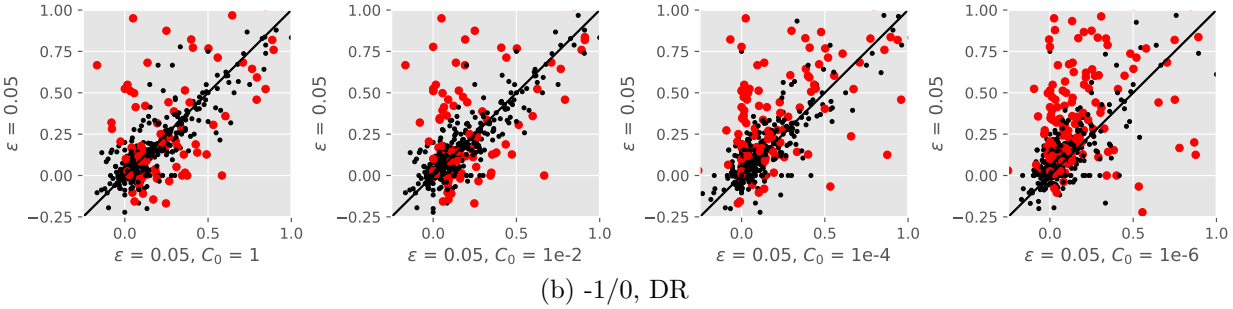
↓ vs →	01	01b	-10	-10b
01	-	42 / 55	65 / 88	33 / 91
01b	55 / 42	-	67 / 62	38 / 63
-10	88 / 65	62 / 67	-	29 / 58
-10b	91 / 33	63 / 38	58 / 29	-

↓ vs →	01	01b	-10	-10b
01	-	78 / 17	44 / 74	71 / 39
01b	17 / 78	-	23 / 102	25 / 58
-10	74 / 44	102 / 23	-	71 / 21
-10b	39 / 71	58 / 25	21 / 71	-

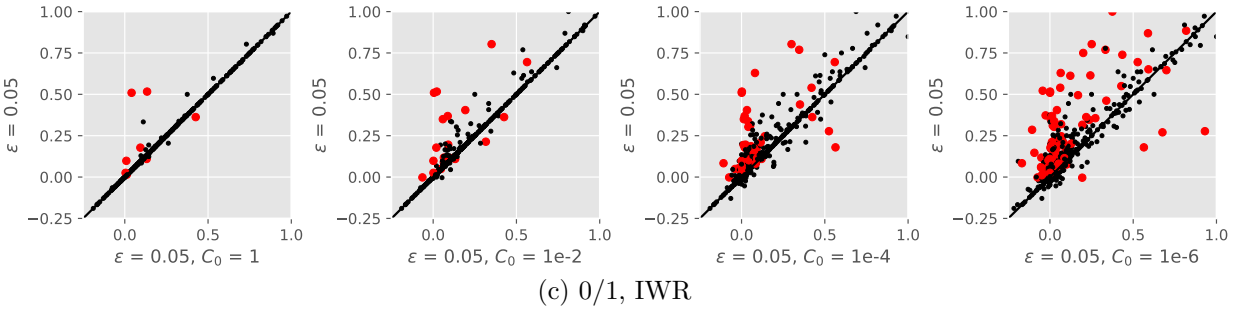
Table 7: Detailed win/loss version of Table 5 with significant wins/losses of row vs column: impact of encoding/baseline for Greedy (left) and Cover-nu (right), with DR, and the remaining hyperparameters optimized.



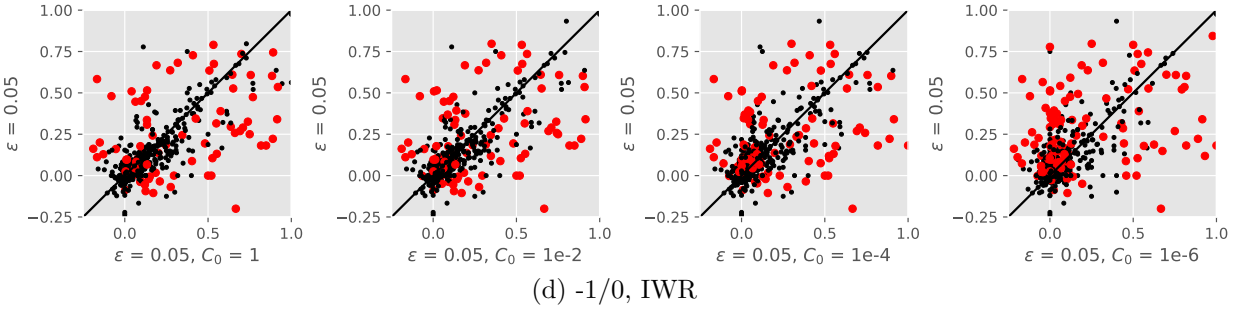
(a) 0/1, DR



(b) -1/0, DR



(c) 0/1, IWR



(d) -1/0, IWR

Figure 5: Improvements to ϵ-greedy from our active learning strategy. Note that active ϵ-greedy shifts observed losses to $[0, 1]$, causing a different behavior for -1/0. The IWR implementation described in Appendix A.1 still manages to often outperform ϵ-greedy, despite only providing an approximation to Algorithm 5.

↓ vs →	G	C-nu	B	B-g	ϵG	C-u	A
G	-	41 / 51	85 / 12	87 / 14	124 / 10	218 / 17	114 / 30
C-nu	51 / 41	-	96 / 13	93 / 14	154 / 4	250 / 1	112 / 5
B	12 / 85	13 / 96	-	6 / 16	81 / 23	177 / 15	82 / 44
B-g	14 / 87	14 / 93	16 / 6	-	90 / 16	183 / 15	88 / 37
ϵG	10 / 124	4 / 154	23 / 81	16 / 90	-	133 / 28	59 / 86
C-u	17 / 218	1 / 250	15 / 177	15 / 183	28 / 133	-	37 / 166
A	30 / 114	5 / 112	44 / 82	37 / 88	86 / 59	166 / 37	-

(a) -1/0 encoding; baseline, reduction and other hyperparameters optimized

↓ vs →	G-iwr	G-dr	C-nu	B	B-g	ϵG	C-u	A
G-iwr	-	15 / 59	66 / 76	134 / 35	131 / 32	128 / 33	247 / 24	124 / 38
G-dr	59 / 15	-	75 / 44	157 / 11	157 / 14	146 / 8	246 / 10	141 / 20
C-nu	76 / 66	44 / 75	-	137 / 46	136 / 45	131 / 21	265 / 4	126 / 15
B	35 / 134	11 / 157	46 / 137	-	12 / 15	76 / 83	190 / 17	90 / 76
B-g	32 / 131	14 / 157	45 / 136	15 / 12	-	80 / 81	196 / 17	93 / 69
ϵG	33 / 128	8 / 146	21 / 131	83 / 76	81 / 80	-	194 / 42	92 / 81
C-u	24 / 247	10 / 246	4 / 265	17 / 190	17 / 196	42 / 194	-	29 / 204
A	38 / 124	20 / 141	15 / 126	76 / 90	69 / 93	81 / 92	204 / 29	-

(b) -1/0 encoding; baseline, reduction and other hyperparameters fixed by Table 2

↓ vs →	G	C-nu	B	B-g	ϵG	C-u	A
G	-	53 / 9	129 / 12	108 / 16	159 / 2	261 / 3	118 / 15
C-nu	9 / 53	-	99 / 22	81 / 26	132 / 14	263 / 2	101 / 23
B	12 / 129	22 / 99	-	5 / 23	79 / 35	202 / 4	76 / 49
B-g	16 / 108	26 / 81	23 / 5	-	90 / 25	222 / 3	88 / 34
ϵG	2 / 159	14 / 132	35 / 79	25 / 90	-	171 / 16	63 / 90
C-u	3 / 261	2 / 263	4 / 202	3 / 222	16 / 171	-	4 / 231
A	15 / 118	23 / 101	49 / 76	34 / 88	90 / 63	231 / 4	-

(c) 0/1 encoding; baseline, reduction and other hyperparameters optimized

↓ vs →	G-iwr	G-dr	C-nu	B	B-g	ϵG	C-u	A
G-iwr	-	22 / 73	72 / 71	126 / 69	119 / 71	137 / 41	238 / 50	131 / 66
G-dr	73 / 22	-	88 / 49	135 / 39	129 / 40	167 / 21	256 / 24	131 / 40
C-nu	71 / 72	49 / 88	-	100 / 51	94 / 61	141 / 35	247 / 16	113 / 48
B	69 / 126	39 / 135	51 / 100	-	9 / 20	118 / 67	218 / 10	94 / 70
B-g	71 / 119	40 / 129	61 / 94	20 / 9	-	120 / 64	225 / 9	97 / 59
ϵG	41 / 137	21 / 167	35 / 141	67 / 118	64 / 120	-	184 / 46	97 / 118
C-u	50 / 238	24 / 256	16 / 247	10 / 218	9 / 225	46 / 184	-	21 / 230
A	66 / 131	40 / 131	48 / 113	70 / 94	59 / 97	118 / 97	230 / 21	-

(d) 0/1 encoding; baseline, reduction and other hyperparameters fixed by Table 2

Table 8: Each (row, column) entry shows the number of statistically significant wins / losses for row against column. Tables (a-b) are more detailed versions of Table 3, with encoding fixed to -1/0. Tables (c-d) are similar, but with encoding fixed to 0/1. The learning rate is always optimized per dataset, other hyperparameters are optimized differently in each table.

↓ vs →	G	C-nu	B	B-g	ϵG	C-u	A
G	-	32 / 27	112 / 8	102 / 11	136 / 3	245 / 4	127 / 6
C-nu	27 / 32	-	109 / 13	94 / 11	144 / 7	254 / 2	124 / 4
B	8 / 112	13 / 109	-	6 / 17	70 / 23	190 / 7	82 / 15
B-g	11 / 102	11 / 94	17 / 6	-	87 / 19	197 / 6	92 / 11
ϵG	3 / 136	7 / 144	23 / 70	19 / 87	-	148 / 19	68 / 61
C-u	4 / 245	2 / 254	7 / 190	6 / 197	19 / 148	-	29 / 166
A	6 / 127	4 / 124	15 / 82	11 / 92	61 / 68	166 / 29	-

(a) encoding, baseline, reduction and hyperparameters optimized

↓ vs →	G-iwr	G-dr	C-nu	B	B-g	ϵG	C-u	A
G-iwr	-	10 / 25	69 / 20	142 / 15	135 / 15	120 / 6	278 / 11	154 / 13
G-dr	25 / 10	-	69 / 14	135 / 9	130 / 9	132 / 4	275 / 2	139 / 4
C-nu	20 / 69	14 / 69	-	98 / 26	91 / 28	101 / 21	273 / 1	129 / 7
B	15 / 142	9 / 135	26 / 98	-	7 / 8	68 / 56	222 / 2	98 / 28
B-g	15 / 135	9 / 130	28 / 91	8 / 7	-	70 / 58	221 / 3	103 / 27
ϵG	6 / 120	4 / 132	21 / 101	56 / 68	58 / 70	-	212 / 21	107 / 54
C-u	11 / 278	2 / 275	1 / 273	2 / 222	3 / 221	21 / 212	-	26 / 194
A	13 / 154	4 / 139	7 / 129	28 / 98	27 / 103	54 / 107	194 / 26	-

(b) encoding and baseline optimized; reduction and other hyperparameters fixed by Table 2

↓ vs →	G-iwr	G-dr	C-nu	B	B-g	ϵG	C-u	A
G-iwr	-	15 / 59	66 / 76	131 / 42	129 / 45	128 / 33	247 / 24	124 / 38
G-dr	59 / 15	-	75 / 44	160 / 19	153 / 20	146 / 8	246 / 10	141 / 20
C-nu	76 / 66	44 / 75	-	128 / 43	121 / 42	131 / 21	265 / 4	126 / 15
B	42 / 131	19 / 160	43 / 128	-	9 / 20	94 / 85	196 / 11	94 / 70
B-g	45 / 129	20 / 153	42 / 121	20 / 9	-	94 / 71	194 / 9	97 / 59
ϵG	33 / 128	8 / 146	21 / 131	85 / 94	71 / 94	-	194 / 42	92 / 81
C-u	24 / 247	10 / 246	4 / 265	11 / 196	9 / 194	42 / 194	-	29 / 204
A	38 / 124	20 / 141	15 / 126	70 / 94	59 / 97	81 / 92	204 / 29	-

(c) encoding, baseline, reduction and hyperparameters fixed by Table 2

Table 9: Each (row, column) entry shows the number of statistically significant wins / losses for row against column. Similar to Table 3 and Table 8, but with different choices of encoding, either fixed by Table 2 or optimized per dataset. The learning rate is always optimized per dataset, other hyperparameters are optimized differently in each table.