Active Learning for Cost-Sensitive Classification

Akshay Krishnamurthy *[1], Alekh Agarwal †[2], Tzu-Kuo Huang ‡[3], Hal Daumé III §[4], and John Langford ¶[2]

[1]University of Massachusetts, Amherst, MA
[2]Microsoft Research, New York, NY
[3]Uber Advanced Technology Center, Pittsburgh, PA
[4]University of Maryland, College Park, MD

March 6, 2017

Abstract

We design an active learning algorithm for cost-sensitive multiclass classification: problems where different errors have different costs. Our algorithm, COAL, makes predictions by regressing on each label's cost and predicting the smallest. On a new example, it uses a set of regressors that perform well on past data to estimate possible costs for each label. It queries only the labels that *could be* the best, ignoring the sure losers. We prove COAL can be efficiently implemented for any regression family that admits squared loss optimization; it also enjoys strong guarantees with respect to predictive performance and labeling effort. We empirically compare COAL to passive learning, showing significant improvements in labeling effort and test cost.

1 Introduction

The field of active learning studies how to efficiently elicit relevant information so learning algorithms can make good decisions. Almost all active learning algorithms are designed for binary classification problems, leading to the natural question: How can active learning address more complex prediction problems? Multiclass and importance-weighted classification require only minor modifications but we know of no active learning algorithms that enjoy theoretical guarantees for more complex problems.

One such problem is cost-sensitive multiclass classification (CSMC). In CSMC with K classes, passive learners receive input examples x and cost vectors $c \in \mathbb{R}^K$, where $c(y)$ is the cost of predicting label y on x.[1] A natural design for an *active* CSMC learner then is to adaptively query the costs of only a (possibly empty) subset of labels on each x. Since measuring label complexity is more nuanced in CSMC (e.g., is it more

*akshay@cs.umass.edu
†alekha@microsoft.com
‡tkhuang@protonmail.com
§hal@umiacs.umd.edu
¶jcl@microsoft.com

[1] Cost here refers to prediction cost and not labeling effort or the cost of acquiring different labels.

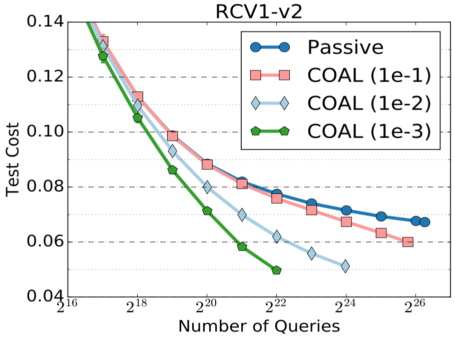


Figure 1: Empirical evaluation of COAL on Reuters text categorization dataset. Active learning achieves *better* test cost than passive, with a factor of 16 fewer queries.

expensive to query three costs on a single example or one cost on three examples?), we track both the number of examples for which at least one cost is queried, along with the total number of queries issued. The first corresponds to a fixed human effort for inspecting x. The second captures the additional effort for judging the cost of each prediction, which depends on the number of labels queried. (By querying a label, we mean querying the cost of that label given an example.)

In this setup, we develop a new active learning algorithm for CSMC called Cost Overlapped Active Learning (COAL). COAL assumes access to a set of regression functions, and, when processing an example x, it uses the functions with good past performance to compute the range of possible costs that each label might take. Naturally, COAL only queries labels with large cost range, but furthermore, it only queries y's that could possibly have the smallest cost, avoiding the uncertain, but surely suboptimal labels. The key algorithmic innovation is an efficient way to compute the cost range realized by good regressors. This computation, and COAL as a whole, only requires that the regression set admits efficient squared loss optimization, in contrast with prior algorithms that require 0/1 loss optimization [6, 16].

Among our results, we prove that when processing n (unlabeled) examples with K classes and N regressors,

1. The algorithm needs to solve $\mathcal{O}(Kn^2 \log n)$ regression problems over the function class (Cor. 2), which can be done in polynomial time for convex regression sets.
2. With no assumptions on the noise in the problem, the algorithm achieves generalization error $\tilde{\mathcal{O}}(\sqrt{K \ln N/n})$ and requests $\tilde{\mathcal{O}}(n\theta_2\sqrt{K \ln N})$ costs from $\tilde{\mathcal{O}}(n\theta_1\sqrt{K \ln N})$ examples (Thms. 3 and 5) where θ_1, θ_2 are the disagreement coefficients (Def. 1)[2]. The worst case offers minimal improvement over passive learning, akin to binary classification.
3. With a favorable noise assumption (As. 2), the algorithm achieves generalization error $\tilde{\mathcal{O}}(K \ln N/n)$ while requesting $\tilde{\mathcal{O}}(Kc^{1/\beta}n^{\beta}\theta_2 \ln N)$ labels from $\tilde{\mathcal{O}}(c^{1/\beta}n^{\beta}\theta_1 K \ln N)$ examples (Cor. 4 and Thm. 6), where $\beta \in (0,1)$ is a safety parameter of the algorithm and c is a constant.

We also discuss some intuitive examples highlighting the benefits of using COAL.

CSMC provides a much more expressive language for success and failure than plain multiclass classification, which allows algorithms to make the tradeoffs necessary for good performance and broadens the potential applications. For example, CSMC can naturally express partial failure in hierarchical classification [27]. Experimentally, we show that on a number of hierarchical classification datasets, COAL

[2] $\tilde{\mathcal{O}}(\cdot)$ suppresses logarithmic dependence on n and K.

substantially outperforms the passive learning baseline with orders of magnitude savings in the labeling effort (see Figure 1 for an example on Reuters text categorization comparing passive learning to COAL).

CSMC also forms the basis of learning to avoid cascading failures in joint prediction tasks [13, 23, 11] like structured prediction and reinforcement learning. As our second application, we consider learning to search algorithms for joint (or structured) prediction [11], which operate by a reduction to CSMC. In this reduction, evaluating the cost of a class often involves a computationally expensive "roll-out," so using an active learning algorithm inside such a (passive) joint prediction method can lead to significant computational savings. We show that using COAL within the AGGRAVATE algorithm [23, 11] reduces the number of roll-outs by a factor of $\frac{1}{4}$ to $\frac{3}{4}$ on several joint prediction tasks.

Related Work. Active learning is a thriving research area with many theoretical and empirical studies. We recommend the survey of Settles [25] for an overview of more empirical research. We focus here on theoretical results.

Theoretical results come in several flavors. Castro and Nowak [9] study active learning for binary classification with non-parametric decision sets, while Balcan et al. [5], Balcan and Long [3] focus on linear representations under distributional assumptions. Additionally, the selective sampling framework from the online learning community derives regret and label complexity bounds for stream-based active learning of linear separators under adversarial assumptions [10, 14, 22, 1].

Our work falls into the framework of *disagreement-based active learning*, which studies general hypothesis spaces typically in an agnostic setup (see Hanneke [16] for an excellent survey). Existing results study binary classification, while our work generalizes to CSMC. The main differences are that our query rule additionally checks the range of predicted costs for a label, and we use a square loss oracle to search the version space.

In contrast, prior work either explicitly enumerates the version space [4, 30] or uses a 0/1 loss *classification* oracle for the search [12, 6, 7, 17]. In most instantiations, the oracle solves an NP-hard problem and so does not directly lead to an efficient active learning algorithm, although practical implementations using heuristics are still quite effective. In contrast, our approach uses a squared-loss *regression* oracle, which can often be implemented efficiently via convex optimization leading to a polynomial time algorithm.

Supervised learning oracles that solve NP-hard optimization problems in the worst case have been used in other problems including contextual bandits [2, 28] and structured prediction [13]. Thus we hope that our work can inspire new algorithms for these settings as well.

2 Problem Setting and Notations

We study a cost-sensitive multiclass classification problem with K classes, where there is an instance space $\mathcal{X}$, a label space $Y = \{1, \dots, K\}$, and a distribution $\mathcal{D}$ supported on $\mathcal{X} \times [0,1]^K$.[3] If $(x, c) \sim \mathcal{D}$, we refer to c as the *cost-vector* where $c(y)$ is the cost of predicting $y \in Y$. A classifier $h : \mathcal{X} \to Y$ has expected cost $\mathbb{E}_{(x,c)\sim\mathcal{D}}[c(h(x))]$ and we aim to find a classifier with minimal expected cost.

Let $\mathcal{G} \triangleq \{g : \mathcal{X} \mapsto [0,1]\}$ denote a set of base regressors and let $\mathcal{F} \triangleq \mathcal{G}^K$ denote a set of vector regressors where the y^{th} coordinate of $f \in \mathcal{F}$ is written as $f(\cdot; y)$. The set of classifiers under consideration is $\mathcal{H} \triangleq \{h_f \mid f \in \mathcal{F}\}$ where each f defines a classifier $h_f : \mathcal{X} \mapsto Y$ by

$$h_f(x) \triangleq \operatorname*{argmin}_y f(x; y). \tag{1}$$

[3] In general, labels just serve as indices for the cost vector in CSMC, and the data distribution is over (x, c) pairs instead of (x, y) pairs as in binary and multiclass classification.

Given a set of examples and queried costs, we often restrict attention to regression functions that predict these costs well, and assess the uncertainty in their predictions given a new example x. For a set of regressors G, we measure uncertainty over possible cost values for y given x with

$$\gamma(x, G) \triangleq \underbrace{c_+(x, G)}_{\triangleq \max_{g \in G} g(x)} - \underbrace{c_-(x, G)}_{\triangleq \min_{g \in G} g(x)} . \tag{2}$$

For vector regressors $F \subset \mathcal{F}$, we define the *cost range* for a label y given x as $\gamma(x, y, F) \triangleq \gamma(x, G_F(y))$ where $G_F(y) \triangleq \{f(\cdot; y) \mid f \in F\}$ is the induced set of base regressors by F for y.

When using a set of regression functions for a classification task, it is natural to assume that the expected costs under $\mathcal{D}$ can be predicted well by the some function in the set. This motivates the following realizability assumption.

Assumption 1 (Realizability)**.** *Define the Bayes-optimal regressor $f^\star$, which has $f^\star(x; y) = \mathbb{E}_c[c(y)|x], \forall x \in \mathcal{X}$ (with $\mathcal{D}(x) > 0$), $y \in Y$. We assume that $f^\star \in \mathcal{F}$.*

$f^\star$ is always well defined although the cost may be noisy, as given by the joint distribution $\mathcal{D}$ on (x, c). In comparison with our assumption, the existence of a zero-cost classifier in $\mathcal{H}$ (which is often assumed in active learning work) is stronger, while the existence of $h_{f^\star}$ in $\mathcal{H}$ is weaker but has not been leveraged in active learning.

In typical settings, the set $\mathcal{G}$ is extremely large, which introduces a computational challenge in managing the version space. To address this challenge, we leverage existing algorithmic research on supervised learning and assume access to a regression oracle for $\mathcal{G}$. Given an importance-weighted dataset $D = \{x_i, c_i, w_i\}_{i=1}^n$ the regression oracle computes

$$\textsc{Oracle}(D) \in \operatorname*{argmin}_{g \in \mathcal{G}} \sum_{i=1}^n w_i (g(x_i) - c_i)^2. \tag{3}$$

In many cases this is a convex problem and can be solved efficiently. In the special case of linear functions, this is just least squares and can be computed in closed form.

To measure the labeling effort, we track the number of examples for which even a single cost is queried as well as the total number of queries. Thus we capture settings where the editorial effort for inspecting an example is high, but each cost requires minimal further effort as well as those where the goal is to just minimize the total number of queries. Formally, we define $Q_i(y)$ to be the indicator that the algorithm queries label y on the i^{th} example and measure

$$L_1 \triangleq \sum_{i=1}^n \bigvee_y Q_i(y), \text{ and } L_2 \triangleq \sum_{i=1}^n \sum_y Q_i(y). \tag{4}$$

3 Cost Overlapped Active Learning

The pseudocode for our algorithm, Cost Overlapped Active Learning (COAL), is given in Algorithm 1. Given an example x, COAL queries the costs of some of the labels y for x. These costs are chosen by (1) computing an approximate version space based on the past data, (2) computing the range of predictions achievable by the version space for each y, and (3) querying each y that could be the best label *and* has substantial uncertainty. We now detail each step.

Algorithm 1: Cost Overlapped Active Learning (COAL)

```
1:  Input: Regressors $\mathcal{G}$, failure probability $\delta \le 1/e$, safety $\beta \in (0,1)$.
2:  Set $\eta_i = 1/\sqrt{i}$, $\kappa = 80$, $\nu_n = \log(2n^2|\mathcal{G}|K/\delta)$.
3:  Set $\Delta_i = \frac{\kappa\epsilon_{i-1}}{i-1}$, $\epsilon_i = \left(\frac{n}{i}\right)^{\beta} \nu_n$.
4:  for $i = 1, 2, \ldots, n$ do
5:      $g_{i,y} \leftarrow \arg\min_{g\in\mathcal{G}} \widehat{R}_i(g;y)$. (See Eq. (5))
6:      Define $f_i \leftarrow \{g_{i,y}\}_{y=1}^{K}$.
7:      $\mathcal{G}_i(y) \leftarrow \{g \in \mathcal{G} \mid \widehat{R}_i(g;y) \le \widehat{R}_i(g_{i,y};y) + \Delta_i\}$.
8:      Receive new example $x$.
9:      for every $y \in Y$ do
10:         $\widehat{c_+}(y) \leftarrow \textsc{MaxCost}((x,y), \Delta_i, \frac{\eta_i}{4\sqrt{3}}, \widehat{R}_i(\cdot;y))$.
11:         $\widehat{c_-}(y) \leftarrow \textsc{MinCost}((x,y), \Delta_i, \frac{\eta_i}{4\sqrt{3}}, \widehat{R}_i(\cdot;y))$.
12:     end for
13:     $Y' \leftarrow \{y \in Y \mid \widehat{c_-}(y) \le \min_{y'} \widehat{c_+}(y')\}$.
14:     if $|Y'| > 1$ then
15:         $Q_i(y) = 1$ if $y \in Y'$ and $\widehat{c_+}(y) - \widehat{c_-}(y) > \eta_i$.
16:     end if
17:     Query costs of each $y$ with $Q_i(y) = 1$.
18: end for
```

To compute an approximate version space we first, for each label y, find the regression function that minimizes the empirical risk for label y, which at round i is:

$$\widehat{R}_i(g;y) = \frac{1}{i-1}\sum_{j=1}^{i-1}(g(x_j) - c_j(y))^2 Q_j(y). \tag{5}$$

Recall that $Q_j(y)$ is the indicator that we query label y on the j^{th} example. Using the square loss is motivated by the realizability assumption and computing the minimizer requires one oracle call. We implicitly construct the version space $\mathcal{G}_i(y)$ in Line 7 as the regressors with low square loss regret to the empirical risk minimizer. The tolerance on this regret is Δ_i at round i, which depends on the safety parameter $\beta \in (0,1)$ in the algorithm. When β is large, the tolerance is also large and the algorithm issues many queries. Conversely when β is small the algorithm is more aggressive. However, for any strictly positive β, the definition of Δ_i ensures that $f^\star(\cdot;y) \in \mathcal{G}_i(y)$ for all i, y.

COAL then computes the maximum and minimum costs predicted by the version space $\mathcal{G}_i(y)$ on the new example x. Since the true expected cost is $f^\star(x;y)$ and $f^\star(\cdot;y) \in \mathcal{G}_i(y)$, these quantities serve as a confidence bound for this value. The computation is done by the MaxCost and MinCost subroutines which produce approximations to $c_+(x, y, \mathcal{G}_i(y))$ and $c_-(x, y, \mathcal{G}_i(y))$ (Eq. (2)) respectively.

Finally, using the predicted costs, COAL issues a set of (possibly zero) queries. The algorithm queries any *non-dominated* label that has a sufficiently large *cost range*, where a label is non-dominated if its estimated minimum cost is smaller than the smallest maximum cost (among all labels) and the cost range is the difference between the label's estimated maximum and minimum costs.

Intuitively, COAL queries the cost of every label which cannot be ruled out as having the smallest cost on x, and where there is sufficient ambiguity about the actual value of the cost. The idea is that labels with little disagreement do not provide much information for further reducing the version space, since by construction

Algorithm 2: MAXCOST

```
Input: $(x, y)$, $\Delta$, $\epsilon$, risk functional $\widehat{R}(\cdot; y)$
$g_{\min} = \text{argmin}_{ginG} \widehat{R}(g; y)$.
$\ell = 0, h = 1, c = 1$.
while $|h - \ell| \geq 2\sqrt{3}\epsilon$ do
    $g_c \leftarrow \text{argmin}_{g \in \mathcal{G}} \widetilde{R}(g, \Delta/\epsilon^2, c; y)$ (see Eq. 6).
    If $g_c \in \mathcal{G}(\Delta; y)$ (see Eq. 7), output $g_c(x) + \epsilon$.
    $(g_l, g_h) \leftarrow \text{BSEARCH}((x, y, c), \epsilon, \Delta, \widehat{R}(\cdot; y))$.
    If $g_h \in \mathcal{G}(4\Delta; y)$, output $g_h(x)$.
    Else $\ell \leftarrow \max\{g_l(x), \ell\}$, $h \leftarrow g_h(x)$, $c \leftarrow \frac{h+\ell}{2}$.
end while
return $c$.
```

all functions would suffer similar loss. Moreover, only the labels that could be the best need to be queried at all, since the cost-sensitive performance of a hypothesis h_f depends only on the label that it predicts to be the best. Hence labels that are dominated or have small cost range need not be queried.

Similar querying strategies were used in prior works on binary and multiclass classification [22, 14, 1], but specialized to linear representations. The key advantage of the linear case is that the set $\mathcal{F}$ (formally, a different set with similar properties) can be maintained in closed form. This further leads to closed form solutions for $c_+(y)$ and $c_-(y)$, so that the algorithms are easily implemented. However, with a general set $\mathcal{G}$ and a regression oracle, computing these confidence intervals is less straightforward. We use the MAXCOST and MINCOST subroutines, and discuss this aspect of our algorithm next.

3.1 Efficient Computation of Cost Range

In this section, we describe the subroutines MAXCOST and MINCOST which use the oracle to compute approximations to the maximum and minimum cost on label y realized by $\mathcal{G}_i(y)$, the current version space (Eq. (2)).

Describing the algorithm requires some additional notation. Given the empirical risk functional $\widehat{R}(g; y)$ over a set of examples (we suppress the subscript as the number of examples is fixed here), we define a weighted risk functional incorporating a fresh unlabeled example x as

$$\widetilde{R}(g, w, c; y) = \widehat{R}(g; y) + w(g(x) - c)^2. \tag{6}$$

Finding $\text{argmin}_g \widetilde{R}(g, w, c; y)$ involves a single oracle call. We also define a set of near-optimal regressors

$$\mathcal{G}(\Delta; y) = \Big\{ g \in \mathcal{G} \mid \widehat{R}(g; y) - \min_{g'} \widehat{R}(g'; y) \leq \Delta \Big\}. \tag{7}$$

Thus at round i, the set $\mathcal{G}_i(y)$ in COAL is equivalent to $\mathcal{G}(\Delta_i; y)$, although we will use different radii here.

The algorithm for the maximum cost approximation, displayed in Algorithm 2, is based on a form of binary search. (The minimum cost approximation is analogous.) The key idea is to solve a sequence of carefully designed regression problems involving the data accumulated so far along with the (x, y) pair in question with different weights.

When invoked with a radius parameter Δ, the algorithm maintains an interval $[\ell, h]$ that contains $c_+(x, y, \mathcal{G}(\Delta; y))$ and uses a binary search to refine the interval. Using a fixed cost c and starting with

Algorithm 3: BinarySearch(BSearch)

```
Input: $(x, y, c)$, $\epsilon$, $\Delta$, risk functional $\widehat{R}(\cdot; y)$.
$w_{1,\ell} = 0, w_{1,h} = \Delta/\epsilon^2, t = 1$.
while $|w_{t,\ell} - w_{t,h}| \geq 2\Delta$ do
    $w_t \leftarrow \frac{w_{t,\ell}+w_{t,h}}{2}$, $g_t = \mathrm{argmin}_{g \in \mathcal{G}} \widetilde{R}(g, w_t, c; y)$.
    If $g_t \in \mathcal{G}(\Delta; y)$, $w_{t+1,\ell} \leftarrow w_t, w_{t+1,h} \leftarrow w_{t,h}$.
    Else $w_{t+1,\ell} \leftarrow w_{t,\ell}$, $w_{t+1,h} \leftarrow w_t$.
    $t \leftarrow t + 1$.
end while
return $g_\ell = \mathrm{argmin}_g \widetilde{R}(g, w_{t,\ell}, c; y)$, and $g_h = \mathrm{argmin}_g \widetilde{R}(g, w_{t,h}, c; y)$.
```

some initial weight w, at each iteration, the binary search computes $\mathrm{argmin}_g \widetilde{R}(g, w, c; y)$ and verifies if the resulting regressor belongs to $\mathcal{G}(\Delta; y)$. If it does, it increases w, and otherwise it shrinks w. Once a termination criteria is reached, the BinarySearch routine outputs two regressors (g_ℓ, g_h) that provide new upper and lower bounds on $c_+(x, y, \mathcal{G}(\Delta; y))$.

The MaxCost routine terminates and outputs $g_h(x)$ if it has reasonable empirical regret. Otherwise, it updates parameters for the next binary search based on $g_\ell(x), g_h(x)$.

Our main algorithmic result guarantees that this procedure produces an adequate approximation to $c_+(x, y, \mathcal{G}(\Delta; y))$ without requiring too many oracle calls.

Theorem 1. *For any (x, y), Δ, and ϵ, the* MaxCost *algorithm outputs $\hat{c}$ satisfying*

$$c_+(x, y, \mathcal{G}(\Delta; y)) \leq \hat{c} \leq c_+(x, y, \mathcal{G}(4\Delta; y)) + \sqrt{3}\epsilon.$$

Further, the algorithm uses $\mathcal{O}(\epsilon^{-2}\log(1/\epsilon))$ oracle calls.

An immediate consequence of the theorem is a bound on the oracle complexity of COAL.

Corollary 2. *Over the course of n examples,* COAL *makes $O(Kn^2 \log(n))$ calls to the square loss oracle.*

Thus COAL can be implemented in polynomial time for any set $\mathcal{G}$ that admits efficient square loss optimization. However, in practice, the number of oracle calls and the oracle itself are too computationally demanding to scale to larger problems. Our implementation alleviates this with an alternative heuristic approximation based on a sensitivity analysis of the oracle, which we detail in Section 6.

4 Generalization Analysis

In this section, we derive generalization guarantees for COAL. Our analysis assumes that the regressor set $\mathcal{G}$ is large, but finite. We study two different settings: one with minimal assumptions and one low-noise setting.

Our low-noise assumption is related to the Massart noise condition [21], which in binary classification posits that the Bayes optimal predictor is bounded away from $1/2$ for all x. Our condition generalizes this to CSMC and posits that, the expected cost of the best label is separated from the expected cost of all other labels.

Assumption 2. *A distribution $\mathcal{D}$ supported over (x, c) pairs satisfies the* Massart noise condition, *if there exists $\tau > 0$ such that for all x (with $\mathcal{D}(x) > 0$),*

$$f^\star(x; y^\star(x)) \leq \min_{y \neq y^\star(x)} f^\star(x; y) - \tau,$$

where $y^\star(x) = \text{argmin}_y\, f^\star(x; y)$.

The Massart noise condition describes favorable prediction problems that lead to sharper generalization and label complexity bounds for COAL. COAL can also be analyzed under a milder assumption inspired by the Tsybakov noise condition, an analysis that we defer to an extended version.

Our results depend on the noise level in the problem, which we define using the following quantity, given any $\zeta > 0$.

$$P_\zeta \triangleq \Pr_{x \sim \mathcal{D}}[\min_{y \neq y^\star(x)} f^\star(x; y) - f^\star(x; y^\star(x)) \leq \zeta]. \tag{8}$$

P_ζ describes the probability that the expected cost of the best label, which is $y^\star(x)$, is close to the expected cost of the second best label. When P_ζ is small for large ζ the labels are well-separated so learning is easier. For instance, under a Massart condition $P_\zeta = 0$ for all $\zeta \leq \tau$.

We now state our generalization guarantee.

Theorem 3. *For any $\delta < 1/e$, for all $i \in [n]$, with probability at least $1 - \delta$, we have*

$$\mathbb{E}_{x,c}[c(h_{f_{i+1}}(x)) - c(h_{f^\star}(x))] \leq \min_{\zeta > 0} \left\{ \zeta P_\zeta + \frac{2\kappa K \nu_n}{\zeta i} \right\},$$

where $\kappa = 80$, $\nu_n = \log\left(\frac{2n^2|\mathcal{G}|K}{\delta}\right)$, f_i is as defined in Line 6 of Algorithm 1, and h_{f_i} is defined in Equation (1).

In the worst case, we bound P_ζ by 1 and optimize for ζ to obtain an $\tilde{\mathcal{O}}(\sqrt{K \log(|\mathcal{G}|/\delta)/i})$ bound after i samples. To compare, the standard generalization bound is $\tilde{\mathcal{O}}(\sqrt{\log(|\mathcal{F}|/\delta)/i})$ [20], which agrees with our bound since $|\mathcal{F}| = |\mathcal{G}|^K$ in our case.

However, since the bound captures the difficulty of the CSMC problem as measured by P_ζ, we can obtain a sharper result under Assumption 2 by setting $\zeta = \tau$.

Corollary 4. *Under Assumption 2, for any $\delta < 1/e$, for all $i \in [n]$, with probability at least $1 - \delta$, we have*

$$\mathbb{E}_{x,c}[c(h_{f_{i+1}}(x)) - c(h_{f^\star}(x))] \leq \frac{2\kappa K \nu_n}{i\tau}.$$

Thus, Massart-type conditions lead to a faster $\tilde{\mathcal{O}}(1/n)$ convergence rate. This agrees with the literature on active learning for classification [21] and can be viewed as a generalization to CSMC. Importantly, both generalization bounds recover the optimal rates and are independent of the safety parameter β.

5 Label Complexity Analysis

Without distributional assumptions, the label complexity of COAL can be $\mathcal{O}(n)$, just as in the binary classification case, since there may always be confusing labels that force querying. In line with prior work, we introduce two **disagreement coefficients** that characterize favorable distributional properties. We first define a set of good classifiers, the *cost-sensitive regret* ball:

$$\mathcal{F}_{\text{csr}}(r) = \left\{ f \in \mathcal{F} \;\middle|\; \mathbb{E}\left[c(h_f(x)) - c(h_{f^\star}(x))\right] \le r \right\}.$$

We may now define the disagreement coefficients.

Definition 1 (Disagreement coefficients)**.** *Define*

$$\begin{aligned} \gamma_r(x,y) &= \gamma(x,y,\mathcal{F}_{csr}(r)), \quad \textit{and} \\ \textit{DIS}(r,y) &= \{x \mid \exists f, f' \in \mathcal{F}_{csr}(r), h_f(x) = y \neq h_{f'}(x)\}. \end{aligned}$$

Then the disagreement coefficients are defined as:

$$\begin{aligned} \theta_1 &\triangleq \sup_{\eta_1, r>0} \frac{\eta_1}{r} \mathbb{P}\left(\exists y \mid \gamma_r(x,y) > \eta_1 \wedge x \in \textit{DIS}(r,y)\right) \\ \theta_2 &\triangleq \sup_{\eta_1, r>0} \frac{\eta_1}{r} \sum_y \mathbb{P}\left(\gamma_r(x,y) > \eta_1 \wedge x \in \textit{DIS}(r,y)\right). \end{aligned}$$

Intuitively, the conditions in both coefficients correspond to the checks on the *domination* and *cost range* of a label in Lines 13 and 15 of Algorithm 1. Specifically, when $x \in \text{DIS}(r,y)$, there is confusion about whether y is the optimal label or not, and hence y is not dominated. The condition on $\gamma_r(x,y)$ additionally captures the fact that a small cost range provides little information, even when y is non-dominated. Collectively, the coefficients capture the probability of an example x where the good classifiers disagree substantially on x in both predicted costs and labels. Importantly, the notion of good classifiers is via the algorithm-independent set $\mathcal{F}_{\text{csr}}(r)$, and is only a property of $\mathcal{F}$ and $\mathcal{D}$.

The definition is a natural adaptation from binary classification [16], where a similar disagreement region to $\text{DIS}(r,y)$ is used. Our definition asks for confusion about the optimality of a specific label y, which provides more detailed information about the cost-structure than simply asking for any confusion among the good classifiers. The $1/r$ scaling leads to bounded coefficients in many examples [16], and we also scale by the cost range parameter η_1, so that the favorable settings for active learning can be concisely expressed as having θ_1, θ_2 bounded, as opposed to a complex function of η_1.

The next two results bound the labeling effort (Def. (4)) in the high noise and low noise cases respectively. The low noise assumption enables a significantly sharper bound.

Theorem 5. *With probability at least $1 - 2\delta$, the label complexity of the algorithm over n examples is bounded by,*

$$\begin{aligned} L_1 &= \mathcal{O}\left((25)^{1/\beta}\left(n\theta_1\sqrt{K\nu_n} + \log(1/\delta)\right)\right) \\ L_2 &= \mathcal{O}\left((25)^{1/\beta}\left(n\theta_2\sqrt{K\nu_n} + K\log(1/\delta)\right)\right), \end{aligned}$$

where $\nu_n = \log\left(\frac{2n^2|\mathcal{G}|K}{\delta}\right)$.

Theorem 6. *Assume the Massart noise condition holds. With probability at least $1-2\delta$ the label complexity of the algorithm over n examples is at most,*

$$L_1 = \mathcal{O}\left(\frac{25^{1/\beta}}{\tau^2}\left(n^\beta K\log(n)\nu_n\theta_1 + \log(1/\delta)\right)\right)$$
$$L_2 = \mathcal{O}\left(\frac{25^{1/\beta}K}{\tau^2}\left(n^\beta\log(n)\nu_n\left[K\theta_1+\theta_2\right] + \log(1/\delta)\right)\right).$$

In the high-noise case, the bounds scales with $n\theta$ for the respective coefficients. This agrees with results in binary classification, where at best constant-factor savings over passive learning are possible, when the disagreement coefficient is small. On the other hand, in the low noise case, the label complexity scales as $\tilde{\mathcal{O}}(c^{1/\beta}n^\beta\theta/\tau^2)$ for the appropriate coefficient, which is a polynomial improvement over passive learning. However, the constant in the label complexity scales exponentially with $1/\beta$ so β should not be chosen to be arbitrarily small. The influence of β in our bound arises from using shrinking radii to ensure bad regressors do not influence the query rule. However we do believe that sharper bounds are possible.

Note that θ_2 can be much smaller than $K\theta_1$, as demonstrated through an example in the next section. In such cases, only a few labels are ever queried and the L_2 bound in the high noise case is more interesting. Unfortunately, under Massart-noise, the L_2 bound depends directly on $K\theta_1$ along with θ_2, so that we do not benefit when $\theta_2 \ll K\theta_1$. If we allow the η_i parameter to depend on the noise level τ, we can obtain a better bound solely depending on θ_2 for L_2. However, we prefer to use the more robust choice $1/\sqrt{i}$ which still allows COAL to partially *adapt* to low-noise and achieves low label complexity.

Unfortunately, translating our bounds to binary classification reveals suboptimality here. In particular, under Massart noise and bounded coefficients, the label complexity is typically $\log(n)/\tau^2$ which contrasts with our n^β/τ^2 rate. Information-theoretically, the logarithmic rate is possible in CSMC, but it remains open whether an efficient algorithm can achieve it.

Our loss in rate arises from setting $\beta > 0$, but when $\beta = 0$, sub-optimal regressors will leave the version space at some round $i < n$, at which point we stop accumulating evidence against them. Since with $\beta = 0$ the radius Δ_i is non-decreasing, these regressors may re-enter the space at some later round and cause us to issue more queries. In binary classification, ideas based on hallucinating labels for unqueried examples address this issue [12], but this technique does not seem applicable here since the only safe choice of hallucinated cost that avoids eliminating f^* appears to be $f^*(x; y)$, which is naturally unknown. Our solution uses $\beta > 0$ to ensure a shrinking radius. However, in order to avoid eliminating $f^\star$, the initial radius Δ_1 must be larger than is required for standard concentration arguments, so the algorithm is somewhat conservative.

5.1 Some Examples

We now describe two examples to give more intuition for COAL and our label complexity bounds.

Our first example shows that querying only the non-dominated labels can dramatically reduce the label complexity. Consider a problem under Assumption 2, where the optimal cost is predicted perfectly, the second best cost is τ worse and all the other costs are substantially worse, but with variability in the predictions. Since all classifiers predict the right label, we get $\theta_1 = \theta_2 = 0$, so our label complexity bound is $\mathcal{O}(1)$. More intuitively, since every regressor is always certain of the optimal label and its cost, we actually make zero queries. On the other hand, all of the suboptimal labels have large cost ranges, and hence querying based solely on a cost range criteria leads to a large label complexity.

A related example demonstrates the improvement in our query rule over more naïve approaches where we query either no label or all labels, which is the natural generalization of query rules from multiclass classification [1]. In the above example, if the best and second best labels are confused occasionally θ_1 may be large, but we expect $\theta_2 \ll K\theta_1$ since only the second best label can have $m_r(x, y)$ small. Thus, the L_2 bound for our algorithm is a factor of K smaller than with a naive query rule since COAL only queries the best and second best labels.

6 Experiments

For computational efficiency, we implemented an approximate version of Algorithm 1 using *online optimization*, based on online linear least-squares regression. The algorithm processes the data in one pass, computing an approximate ERM and cost ranges as described below.

The idea is to replace $g_{i,y}$, the ERM, with an approximation $g^o_{i,y}$ obtained by online updates, and compute the minimum and maximum costs via a sensitivity analysis of the online update. Specifically, we define a *sensitivity* value $s(x, c, g^o_{i,y}) \geq 0$, which is the derivative of the prediction on x as a function of the importance weight w, for a fresh example x and cost $c = 0$ or $c = 1$ (for approximating c_- and c_+ respectively). Then we approximate c_- via $g^o_{i,y}(x) - \underline{w}^o \cdot s(x, 0, g^o_{i,y})$ where $\underline{w}^o$ is the largest weight w satisfying

$$w(g^o_{i,y}(x)^2 - (g^o_{i,y}(x) - ws(x, 0, g^o_{i,y}))^2) \leq \Delta_i,$$

where Δ_i is the radius used at round i. We use an analogous technique to approximate the maximum cost. See Appendix A for more details.

6.1 Simulated Active Learning

We performed simulated active learning experiments with three datasets. ImageNet 20 and 40 are sub-trees of the ImageNet hierarchy covering 20 and 40 most frequent classes, where each example has a single zero-cost label and the cost for incorrect labels is the tree-distance to the correct one. The feature vectors are the top layer of the Inception neural network [29]. The third dataset, RCV1-v2 [19], is a multilabel text-categorization dataset, which has 103 topic labels, organized as a tree with similar tree-distance cost structure as the ImageNet data. Some dataset statistics are in Figure 2 (upper right).

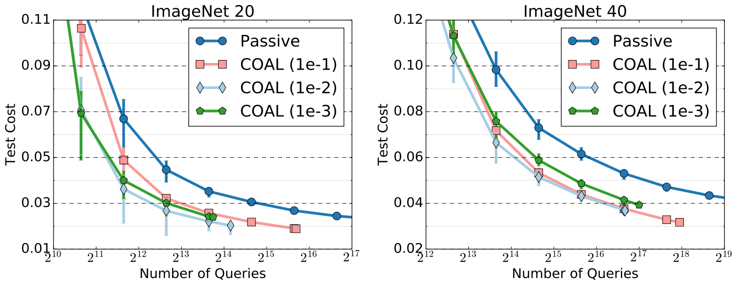
We compare our online version of COAL to passive online learning. We use the cost-sensitive one-against-all (CSOAA) implementation in Vowpal Wabbit[4], which performs online linear regression for each label separately. There are two tuning parameters in our implementation. First, instead of Δ_i, we set the radius of the version space to $\Delta'_i = \frac{\kappa\nu_{i-1}}{i-1}$ (i.e. $\beta = 0$ and the log factor $\nu_i = \log\left(\frac{2(i-1)^2|\mathcal{G}|K}{\delta}\right)$ scales with i) and instead tune the constant κ. This alternate “mellowness" parameter controls how aggressive the query strategy is. The second parameter is the learning rate used by online linear regression[5].

For each parameter setting and each dataset, we make one pass through the training set and check the test cost (which is just the normalized expected cost) of the model every doubling number of queries. We repeat this on 100 random permutations of the training data and plot the results in Figure 2. For each mellowness, we show the results of the best learning rate, which maximizes a notion of AUC that reflects the tradeoff between test cost and number of queries (see Eq. (11) in Appendix A).

Figure 2 shows, for each dataset and mellowness, the number of queries against the median test cost along with bars extending from the 15th to 85th quantile. Overall, COAL achieves a better trade-off between

[4] http://hunch.net/~vw

[5] We use the default online learning algorithm in Vowpal Wabbit, which is a scale-free [24] importance weight invariant [18] form of AdaGrad [15].



	K	n	feat	ℓ
ImageNet 20	20	38k	6k	-
ImageNet 40	40	71k	6k	-
RCV1-v2	103	781k	47k	-
POS	45	38k	40k	24
NER	9	15k	15k	14
NER-wiki	9	132k	89k	25

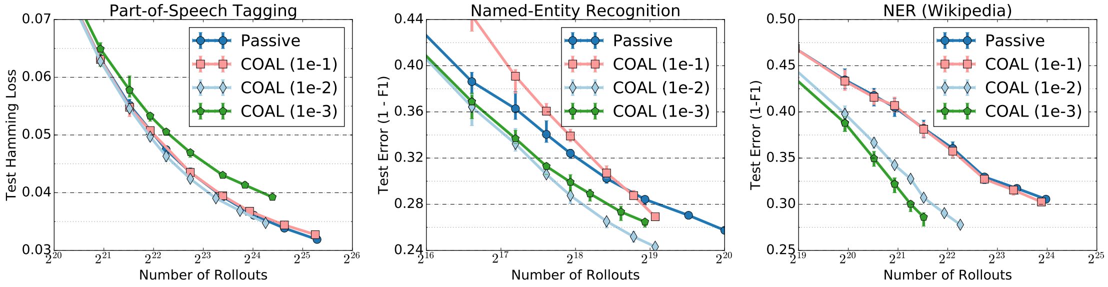


Figure 2: Experiments with COAL. Top row shows test cost vs. number of queries for simulated active learning experiments. Bottom row shows accuracy vs. number of rollouts for active and passive learning as the CSMC algorithm in learning-to-search. Upper-right shows dataset statistics for both experimental settings (ℓ is the average sequence length for the sequence labeling experiments).

performance and queries. With proper mellowness parameter, active learning achieve similar test cost as passive learning with a factor of 8 to 32 less queries. On ImageNet 40 and RCV1-v2 (recall Figure 1), active learning achieves *better* test cost with a factor of 16 less queries. On RCV1-v2, COAL queries like passive up to around $256k$ queries, since the data is very sparse, and linear regression has the property that the cost range is maximal when an example has a new unseen feature. Once COAL sees all features a few times, it queries much more efficiently than passive. Note that these plots correspond to the label complexity L_2, with similar results for L_1 in Appendix A.3.

While not always the best, we recommend using a mellowness setting of 0.01 in practice as it achieves reasonable performance on all three datasets. This is also confirmed by the learning-to-search experiments, which we discuss next.

6.2 Learning to Search

We also experiment with COAL as the base leaner in *learning-to-search* [13, 11], which reduces joint prediction problems to CSMC. In this framework, a joint prediction example defines a search space, where a sequence of decisions are made to generate the structured label. We focus here on sequence labeling tasks, where the input is a sequence of words and the output is a sequence of labels (specifically, parts of speech or named entities).

Learning-to-search solves joint prediction problems by generating the output one label at a time, conditioning the input x on all past decisions. Since mistakes may lead to compounding errors, it is natural to represent the decision space as a CSMC problem, where the classes are the “actions” available (possible labels for a word) and the costs reflect the long term loss of each choice. Intuitively, we should be able to avoid expensive computation of long term loss on decisions like "is 'the' a DETERMINER?" once we are quite sure of the answer. Similar ideas motivate adaptive sampling for structured prediction. [26].

We specifically use AGGRAVATE [23, 11], which runs a learned policy to produce a backbone sequence of labels. For each position in the input sentence, it then considers all possible deviation actions and executes an oracle for the rest of the sequence. The loss on this complete output is used as the cost for the deviating action. Run in this way, AGGRAVATE requires mK roll-outs when the input sentence has m words and each word can take one of K possible labels.

Since each roll-out takes $\mathcal{O}(m)$ time, this can be computationally prohibitive, and hence we use active learning to reduce the number of roll-outs. We use COAL and a passive learning baseline inside AGGRAVATE on three joint prediction datasets (dataset statistics are in Figure 2, upper right). As above, we use several mellowness values and the same AUC criteria to select the best learning rate. The results are in the bottom row of Figure 2, with the black arrow pointing to test cost for our recommended mellowness of 0.01.

Overall, active learning reduces the number of roll-outs required, but the improvements vary on the three datasets. On the Wikipedia data, COAL performs a factor of 4 less rollouts to achieve similar performance to passive learning, and it also achieves substantially better test performance. A similar, but less dramatic, behavior arises on the NER task. On the other hand, COAL offers minimal improvement over passive learning on the POS-tagging task.

7 Discussion

This paper presents a new active learning algorithm for cost-sensitive multiclass classification. The algorithm enjoys strong theoretical guarantees on running time, generalization error, and label complexity, and also outperforms passive baselines both in CSMC and structured prediction.

We close with some intriguing questions:

1. Can we use a square loss oracle in other partial information problems like contextual bandits?
2. Can we avoid the safety parameter to achieve the optimal complexity in the low noise case?

We hope to answer these questions in future work.

A Experimental Details

A.1 Finding Cost Ranges with Online Approximation

Consider the maximum and minimum costs for a fixed label y at round i, both of which may be suppressed. Owing to the monotonicity property of $\hat{R}(g, w, c; y)$ (Lemma 1), an alternative to MinCost and MaxCost is to find

$$\underline{w} := \max\{w \mid \widehat{R}(\underline{g}_{w}) - \widehat{R}(g_{i,y}) \leq \Delta_i\} \tag{9}$$

$$\overline{w} := \max\{w \mid \widehat{R}(\overline{g}_{w}) - \widehat{R}(g_{i,y}) \leq \Delta_i\} \tag{10}$$

and return $\underline{g}_{\underline{w}}(x)$ and $\overline{g}_{\overline{w}}(x)$ as the minimum and maximum costs, where

$$\underline{g}_w \triangleq \arg\min_{g \in \mathcal{G}} \widehat{R}(g) + w(g(x) - 0)^2$$

$$\overline{g}_w \triangleq \arg\min_{g \in \mathcal{G}} \widehat{R}(g) + w(g(x) - 1)^2$$

and $g_{i,y} \triangleq \operatorname{argmin}_{g \in \mathcal{G}} \widehat{R}(g)$ as in Algorithm 1. We use two steps of approximation here. Using the definition of $\overline{g}_w$ and $\underline{g}_w$ we have:

$$\widehat{R}(\underline{g}_w) - \widehat{R}(g_{i,y}) \leq w \cdot g_{i,y}(x)^2 - w \cdot \underline{g}_w(x)^2$$

$$\widehat{R}(\overline{g}_w) - \widehat{R}(g_{i,y}) \leq w \cdot (g_{i,y}(x) - 1)^2 - w \cdot (\overline{g}_w(x) - 1)^2.$$

We use this upper bound in place of $\widehat{R}(g_w) - \widehat{R}(g_{i,y})$ in Eqs. (9) and (10). Second, we replace $g_{i,y}$, $\underline{g}_w$, and $\overline{g}_w$ with approximations obtained by online updates. More specifically, we replace $g_{i,y}$ with $g^o_{i,y}$, the current regressor produced by all online updates so far, and approximate the others by

$$\underline{g}_w(x) \approx g^o_{i,y}(x) - w \cdot s(x, 0, g^o_{i,y})$$

$$\overline{g}_w(x) \approx g^o_{i,y}(x) + w \cdot s(x, 1, g^o_{i,y})$$

where $s(x, y, g^o_{i,y}) \geq 0$ is a *sensitivity* value that approximates the change in prediction on x resulting from an online update to $g^o_{i,y}$ with features x and label y. The computation of this sensitivity value is governed by the actual online update where we compute the derivative of the change in the prediction as a function of the importance weight w for a hypothetical example with cost 0 or cost 1 and the same features. This is possible for essentially all online update rules on importance weighted examples where it corresponds to taking the limit as $w \to 0$ of the change in prediction due to an update divided by w. By inspection this requires only $\mathcal{O}(d)$ time per example, where d is the average number of non-zero features. With these two steps, we obtain approximate minimum and maximum costs using:

$$g^o_{i,y}(x) - \underline{w}^o \cdot s(x, 0, g^o_{i,y})$$

$$g^o_{i,y}(x) + \overline{w}^o \cdot s(x, 1, g^o_{i,y})$$

where

$$\underline{w}^o \ \triangleq \ \max\{w \mid w\left(g^o_{i,y,}(x)^2 - (g^o_{i,y}(x) - w \cdot s(x, 0, g^o_{i,y}))^2\right) \leq \Delta_i\}$$

$$\overline{w}^o \ \triangleq \ \max\{w \mid w\left((g^o_{i,y,}(x) - 1)^2 - (g^o_{i,y}(x) + w \cdot s(x, 1, g^o_{i,y}) - 1)^2\right) \leq \Delta_i\}.$$

	ImageNet 20	ImageNet 40	RCV1-v2	POS	NER	NER-wiki
passive	1	1	0.5	1.0	0.5	0.5
active (10^{-1})	0.05	0.1	0.5	1.0	0.1	0.5
active (10^{-2})	0.05	0.5	0.5	1.0	0.5	0.5
active (10^{-3})	1	10	0.5	10	0.5	0.5

Table 1: Best learning rates

The online update guarantees that $g^o_{i,y}(x) \in [0, 1]$. Since the minimum cost is lower bounded by 0, we have $\underline{w}^o \in \left(0, \frac{g^o_{i,y}(x)}{s(x,0,g^o_{i,y})}\right]$. Finally, because the objective $w(g^o_{i,y}(x))^2 - w(g^o_{i,y}(x) - w \cdot s(x, 0, g^o_{i,y}))^2$ is increasing in w within this range (which can be seen by inspecting the derivative), we can find $\underline{w}^o$ with binary search. Using the same techniques, we also obtain an approximate maximum cost.

It is worth noting that the approximate cost ranges (without the sensitivity trick) are contained in the exact cost ranges because we approximate the difference in squared error by an *upper bound*. Hence, the query rule in this online algorithm should be more aggressive than the query rule in Algorithm 1.

A.2 Choosing the Learning Rate

For all experiments, we show the results obtained by the best learning rate for each mellowness on each dataset. We choose the best learning rate as follows. For each dataset let $\mathrm{perf}(m, l, q, t)$ denote the test performance of the algorithm using mellowness m and learning rate l on the t^{th} permutation of the training data under a query budget of $2^{(q-1)} \cdot 10 \cdot K, q \geq 1$. Let $\mathrm{query}(m, l, q, t)$ denote the number of queries actually made. Note that $\mathrm{query}(m, l, q, t) < 2^{(q-1)} \cdot 10 \cdot K$ if the algorithm runs out of the training data before reaching the q^{th} query budget[6] . To evaluate the trade-off between test performance and number of queries, we define the following performance measure:

$$\mathrm{AUC}(m, l, t) = \frac{1}{2} \sum_{q=1}^{q_{\max}} \Big(\mathrm{perf}(m, l, q + 1, t) + \mathrm{perf}(m, l, q, t)\Big) \cdot \left(\log_2 \frac{\mathrm{query}(m, l, q + 1, t)}{\mathrm{query}(m, l, q, t)}\right), \tag{11}$$

where $q_{\max}$ is the minimum q such that $2^{(q-1)} \cdot 10$ is larger than the size of the training data. This performance measure is the area under the curve of test performance against numbers of queries in $\log_2$ scale. A large value means the test performance quickly improves with the number of queries. The best learning rate for mellowness m is then chosen as

$$l^\star(m) \triangleq \arg\max_l \mathrm{median}_{1 \leq t \leq 100} \quad \mathrm{AUC}(m, l, t).$$

The best learning rates for different datasets and mellowness settings are in Table 1.

A.3 Additional Figures for Simulated Active Learning

In Figure 3, we plot the test error as a function of the number of examples for which at least one query was requested, for each dataset and mellowness parameter. This experimentally corresponds to the L_1 term in our label complexity analysis.

[6] In fact, we check the test performance only in between examples, so $\mathrm{query}(m, l, q, t)$ may be larger than $2^{(q-1)} \cdot 10 \cdot K$ by an additive factor of K, which is negligibly small.

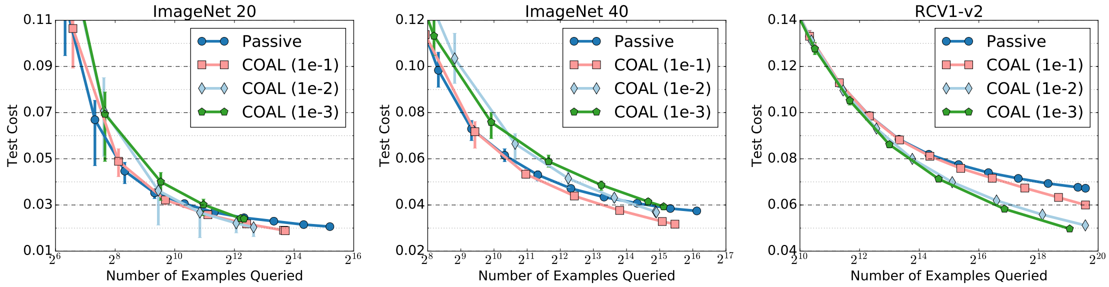


Figure 3: Additional figures for simulated active learning experiments. The plots show the test cost as a function of the number of examples where even a single query was issued.

In comparison with Figure 2 involving the total number of queries, the improvements offered by active learning are slightly less dramatic here. This suggests that our algorithm queries just a few labels for each example, but does end up issuing at least one query on most of the examples. Nevertheless, one can still achieve test cost competitive with passive learning using a factor of 2-16 less labeling effort, as measured by L_1.

B Running time analysis

Throughout this section, fix an x, y pair, an iteration i, as well a radius Δ and an accuracy ϵ. We focus on approximating $c_+(x, \mathcal{G}(\Delta; y))$ (See Eqs. (2) and (7)), approximating the minimum cost is very similar. To simplify notation, we drop dependence on x and y. We recall our earlier notation $\widehat{R}_i(g; y)$ (Eq. (5)), except we drop the dependence on both y and i which are fixed throughout this appendix. We also recall some other important pieces of notation which are accordingly simplified for brevity

$$\begin{aligned}
\widehat{R}(g) &= \hat{\mathbb{E}}[(g(x) - c(y))^2 \mathbb{1}\,(y \text{ queried on } x)] \\
\mathcal{G}(\Delta) &= \{g \in \mathcal{G} : \widehat{R}(g) - \min_g \widehat{R}(g) \le \Delta\} \\
g_{\min} &= \operatorname*{argmin}_{g \in \mathcal{G}} \widehat{R}(g) \\
\widetilde{R}(g, w, c) &= \widehat{R}(g) + w(g(x) - c)^2 \\
c_+(\alpha\Delta) &= \max_{g \in \mathcal{G}(\alpha\Delta)} g(x) \\
c_\star &= c_+(\Delta)
\end{aligned}$$

$\widehat{R}(g)$ is the empirical square loss used to define the set of good regressors $\mathcal{G}(\Delta)$ in the algorithm. The precise form of $\widehat{R}(g)$ does not matter in this section. $g_{\min}$ is the empirical square loss minimizer, which is the center of the ball $\mathcal{G}(\Delta)$. $\widetilde{R}(g, w, c)$ is the empirical square loss with one additional example, with features x, target c, and weight w. This functional is used to define new square loss problems in our algorithm. Our goal is to find a number $\hat{c}$ such that,

$$c_\star \le \hat{c} \le c_+(4\Delta) + \sqrt{3}\epsilon.$$

Finally, let $g_\star$ be any function such that $g_\star(x) = c_\star$ and $\widehat{R}(g_\star) - \widehat{R}(g_{\min}) \le \Delta$. In other words $g_\star$ realizes the maximum cost on example x. Note that $g_\star$ is *not* the same regressor that satisfies the realizability condition.

We start the running time analysis with several lemmas that characterize the behavior of various components of the algorithm.

An important structure to the square loss problem is a monotonicity property of both the risk functional and the predictions.

Lemma 1. *For any c and for $w' \ge w \ge 0$, let $g = \operatorname{argmin}_g \widetilde{R}(g, w, c)$ and $g' = \operatorname{argmin}_g \widetilde{R}(g, w', c)$. Then*

$$\widehat{R}(g') \ge \widehat{R}(g) \text{ and } (g'(x) - c)^2 \le (g(x) - c)^2.$$

Proof. By the definitions,

$$\begin{aligned} \widehat{R}(g') + w'(g'(x) - c)^2 &= \widetilde{R}(g', w', c) \\ &\le \widehat{R}(g) + w'(g(x) - c)^2 \\ &= \widehat{R}(g) + w(g(x) - c)^2 + (w' - w)(g(x) - c)^2 \\ &\le \widehat{R}(g') + w(g'(x) - c)^2 + (w' - w)(g(x) - c)^2. \end{aligned}$$

Rearranging shows that

$$(w' - w)(g'(x) - c)^2 \le (w' - w)(g(x) - c)^2.$$

Since $w' \ge w$, we must have $(g'(x) - c)^2 \le (g(x) - c)^2$, which is the second claim. For the first claim, the definition of g gives

$$\widehat{R}(g) + w(g(x) - c)^2 \le \widehat{R}(g') + w(g'(x) - c)^2$$

Rearranging this inequality gives,

$$\widehat{R}(g') - \widehat{R}(g) \ge w((g(x) - c)^2 - (g'(x) - c)^2) \ge 0. \qquad \square$$

The next critical lemma shows that the termination condition in Line 6 of MaxCost meets the accuracy guarantee.

Lemma 2. *If $c \ge c_\star, w \ge \Delta/\epsilon^2$ and $g = \operatorname{argmin}_g \widetilde{R}(g, w, c)$ then $g(x) \ge c_\star - \epsilon$. Further, if $g \in \mathcal{G}(\Delta)$, then $g(x) \le c_\star$.*

Proof. The second claim is straightforward by the definition of $c_\star$.

For the first claim, we work to establish a contradiction. Suppose that $g(x) < c_\star - \epsilon$. By the facts that g is the minimizer of $\widetilde{R}(g, w, c)$, $g_{\min}$ is the minimizer of $\widehat{R}(g)$, and $c \ge c_\star$, we have

$$w(c - (c_\star - \epsilon))^2 < w(c - g(x))^2 \le \widetilde{R}(g, w, c) - \widehat{R}(g_{\min}) \le \widetilde{R}(g_\star, w, c) - \widehat{R}(g_{\min}) \le \Delta + w(c - c_\star)^2.$$

We may further lower bound (again using $c \ge c_\star$),

$$(c - c_\star + \epsilon)^2 = (c - c_\star)^2 + 2(c - c_\star)\epsilon + \epsilon^2 \ge (c - c_\star)^2 + \epsilon^2.$$

Re-arranging proves that

$$w < \Delta/\epsilon^2.$$

The contrapositive is that if $w \ge \Delta/\epsilon^2$, then we must have $g(x) \ge c_\star - \epsilon$, which is the desired claim. $\square$

The next lemma is the main result for the BINARYSEARCH subroutine.

Lemma 3. *Suppose we invoke the subroutine* BINARYSEARCH *with parameters* ϵ *and* Δ. *Then it terminates in polynomial time with* $O(\log_2(1/(\epsilon^2)))$ *oracle calls. The algorithm outputs two regressors* (g_ℓ, g_h) *and if* $c \geq c_\star$ *is passed as input then* $c_\star \in [g_\ell(x), g_h(x)]$. *If additionally,* $g_h \notin \mathcal{G}(4\Delta)$ *then* $c_\star \leq (g_\ell(x) + g_h(x))/2$.

Proof. The logarithmic running time is fairly straightforward since in each iteration the algorithm halves the interval, has initial interval of size Δ/ϵ^2 and terminates when the interval is smaller than 2Δ. Thus for $T \geq \log_2(1/(2\epsilon^2))$ the interval has size at most

$$2^{-T}(\Delta/\epsilon^2) \leq 2^{-\log_2(1/(2\epsilon^2))}(\Delta/\epsilon^2) = 2\Delta$$

Hence the number of iteration is upper bounded by $\lceil \log_2(1/(2\epsilon^2)) \rceil$.

For the first termination claim, the invariant that we maintain is that for all $t \geq 1$, $g_{t,h} = \mathrm{argmin}_g \widetilde{R}(g, w_{t,h}, c)$ satisfies $g_{t,h}(x) \geq c_\star$ while $g_{t,\ell} = \mathrm{argmin}_g \widetilde{R}(g, w_{t,\ell}, c)$ satisfies $g_{t,\ell}(x) \leq c_\star$.

For $g_{t,h}$, we first establish the base case. Observe that $g_{1,h} = g_c$ (computed in MAXCOST just before the invocation of BINARYSEARCH) and $\widehat{R}(g_c) \geq \widehat{R}(g_{\min}) + \Delta$ by the termination check in Line 6. By construction, in this iteration and in all others, we have that $g_{t,h} \notin \mathcal{G}(\Delta)$, since this is the requirement for updating $w_{t,h}$. But since $g_{t,h}$ minimizes the risk function $\widetilde{R}(g, w_{t,h}, c)$ we get,

$$\widehat{R}(g_{\min}) + \Delta + w_{t,h}(g_{t,h}(x) - c)^2 \leq \widetilde{R}(g_{t,h}, w_{t,h}, c) \leq \widetilde{R}(g_\star, w_{t,h}, c) \leq \widehat{R}(g_{\min}) + \Delta + w_{t,h}(c_\star - c)^2.$$

Since $c \geq c_\star$, this implies that $g_{t,h}(x) \geq c_\star$.

The proof for $g_{t,\ell}$ is simpler, since we only shrink the interval up if we find something in $\mathcal{G}(\Delta)$. By definition of $c_\star$ the cost of $g_{t,\ell}(x)$ for these iterations satisfies $g_{t,\ell}(x) \leq c_\star$.

For the second termination claim we must use the condition $|w_{t,h} - w_{t,\ell}| \leq 2\Delta$ by the termination condition and $g_h \notin \mathcal{G}(4\Delta)$. Let t be the terminal iteration, so $g_\ell = \mathrm{argmin}_g \widetilde{R}(g, w_{t,\ell}, c)$ and analogously for g_h. Assume for the sake of contradiction that $c_\star \geq (g_h(x) + g_\ell(x))/2$. Since $c \geq c_\star$, this implies that

$$\begin{aligned}\widehat{R}(g_{\min}) + 4\Delta + w_{t,h}(g_h(x) - c)^2 \leq \widetilde{R}(g_h, w_{t,h}, c) \leq \widetilde{R}(g_\star, w_{t,h}, c) &\leq \widehat{R}(g_{\min}) + \Delta + w_{t,h}(c - c_\star)^2 \\ &\leq \widehat{R}(g_{\min}) + \Delta + w_{t,h}\left(c - \frac{g_h(x) + g_\ell(x)}{2}\right)^2.\end{aligned}$$

Similarly we have

$$\widehat{R}(g_{\min}) + w_{t,\ell}(g_\ell(x) - c)^2 \leq \widetilde{R}(g_\ell, w_{t,\ell}, c) \leq \widetilde{R}(g_\star, w_{t,\ell}, c) \leq \widehat{R}(g_{\min}) + \Delta + w_{t,\ell}\left(c - \frac{g_h(x) + g_\ell(x)}{2}\right)^2.$$

Adding the two equations gives

$$\begin{aligned}&2\Delta + w_{t,\ell}(c - g_\ell(x))^2 + w_{t,h}(c - g_h(x))^2 \leq (w_{t,h} + w_{t,\ell})\left(c - \frac{g_h(x) + g_\ell(x)}{2}\right)^2 \\ \Rightarrow\ & 2\Delta + w_{t,h}\left[(c - g_\ell(x))^2 + (c - g_h(x))^2\right] \leq 2w_{t,h}\left(c - \frac{g_h(x) + g_\ell(x)}{2}\right)^2 + (w_{t,h} - w_{t,\ell})(c - g_\ell(x))^2 \\ \Rightarrow\ & 2\Delta + w_{t,h}\left[(c - g_\ell(x))^2 + (c - g_h(x))^2\right] \leq 2w_{t,h}\left(c - \frac{g_h(x) + g_\ell(x)}{2}\right)^2 + 2\Delta \quad \text{since } c,\ g_\ell(x) \in [0,1] \\ &\Rightarrow \frac{1}{2}(c - g_\ell(x))^2 + \frac{1}{2}(c - g_h(x))^2 \leq \left(c - \frac{g_h(x) + g_\ell(x)}{2}\right)^2.\end{aligned}$$

The last line is a contradiction since $\mathbb{E}[f(Z)] \geq f(\mathbb{E}[Z])$ for convex f, which can be applied by taking $Z = \text{Unif}(\{g_\ell(x), g_h(x)\})$ and $f(y) = (c-y)^2$. □

The last lemma ensures sufficient progress in the case when $g_h \notin \mathcal{G}(4\Delta)$, which is crucial for the oracle complexity bound.

Lemma 4. *Suppose $c \geq c_\star$ and that there exists $g \in \mathcal{G}(\Delta)$ such that $c - g(x) = \delta$ with $\delta \in [\sqrt{3}\epsilon, 1]$. Then if the output (g_ℓ, g_h) of* BinarySearch *satisfies $\widehat{R}(g_h) - \widehat{R}(g_{\min}) > 4\Delta$, then $g_h(x) \leq c + \delta - \epsilon^2$.*

Proof. We know that we never use a weight larger than Δ/ϵ^2 by the initialization of $w_{1,\ell}, w_{1,h}$. Now suppose that we output g_h such that $\widehat{R}(g_h) > 4\Delta$, which by construction is the minimizer of $\widetilde{R}(g, w, c)$ for some $w \leq \Delta/\epsilon^2$. Then

$$\widehat{R}(g_{\min}) + 4\Delta + w(g_h(x) - c)^2 \leq \widetilde{R}(g_h, w, c) \leq \widetilde{R}(g, w, c) \leq \Delta + w(g(x) - c)^2 = \widehat{R}(g_{\min}) + \Delta + w\delta^2.$$

Rearranging and using the upper bound w gives

$$(g_h(x) - c)^2 \leq \delta^2 - 3\epsilon^2.$$

The condition on δ ensures that the right hand side is non-negative. It is easy to see that $\delta^2 - 3\epsilon^2 \leq (\delta - \epsilon^2/\delta)^2$ simply by expanding the square. Hence we get that

$$|g_h(x) - c| \leq |\delta - \epsilon^2/\delta| \leq \delta - \epsilon^2.$$

We can safely remove the absolute value since we have the condition that $\delta \geq \sqrt{3}\epsilon$, which ensures that $\delta - \epsilon^2/\delta$ is non-negative. Since g_h is the result of an oracle call with weight $w \leq \Delta/\epsilon^2$, either it has $\widehat{R}(g_h) - \widehat{R}(g_{\min}) \leq 4\Delta$, or it must have $g_h(x) \leq c + \delta - \epsilon^2$. □

We are now ready to prove Theorem 1

Proof of Theorem 1. The first step of the proof is to inductively verify that $c \geq c_\star, h \geq c_\star, \ell \leq c_\star$ at all steps in the algorithm execution. These invariants are clearly maintained at the onset of the algorithm. Now suppose they are maintained at the onset of some iteration. If g_c satisfies $\widehat{R}(g_c) \leq \widehat{R}(g_{\min}) + \Delta$, then by Lemma 2 we are done. Otherwise, we obtain two regressors (g_ℓ, g_h) from BinarySearch. For the lower bound, we always have $g_\ell(x) \leq c_\star$ by Lemma 3, which verifies the inductive step for ℓ. For the upper bound to $c_\star$, if $\widehat{R}(g_h) - \widehat{R}(g_{\min}) \leq 4\Delta$ then by Lemma 3, we know that $c_\star \leq g_h(x)$, but we also know that $g_h(x) \leq c_+(4\Delta)$ by the definition, so we are done. The last case is when $\widehat{R}(g_h) > 4\Delta$, but here we may apply the second statement of Lemma 3, which asserts that $c_\star \leq (g_h(x) + g_l(x))/2$. The settings of ℓ, h, c now verify the inductive claim, since $\ell \geq g_l$ implies that $(h + l)/2 \geq (g_h(x) + g_l(x))/2 \geq c_\star$.

This immediately proves the correctness of the algorithm, since the loop stopping condition, along with the invariant, guarantees that $c \geq c_\star \geq \ell$ which means that

$$\hat{c} - c_\star \leq c - \ell = \frac{h - \ell}{2} \leq \sqrt{3}\epsilon.$$

For the iteration complexity, we must apply Lemma 4. In particular, we use the width of the interval $[\ell, h]$ which contains $c_\star$ as a potential function and show that it decreases with every step. Let δ_t denote $h - \ell$, which is the width of the interval before the tth iteration (so $\delta_1 = 1$). Every non-terminal iteration satisfies $c \geq c_\star$. Moreover, for any $t > 1$, we use as the regressor g, the one that achieved the value ℓ used to define c.

This ensures that $g(x) = \ell$. Furthermore, in application of Lemma 4, we set $\delta \triangleq c - g(x) = c - \ell$, which conveniently gives $2\delta = \delta_t = h - \ell$. Recall that we entered the loop at t_{th} iteration, meaning that $\delta_t \geq 2\sqrt{3}\epsilon$ and hence $\delta \in [\sqrt{3}\epsilon, 1]$. Lemma 4 states that either we terminate successfully, or we are guaranteed that $g_h(x) \leq c + \delta - \epsilon^2$. This means that

$$\delta_{t+1} = g_h(x) - \max\{\ell, g_\ell(x)\} \leq c + \delta - \epsilon^2 - \ell = \delta + \delta - \epsilon^2 = \delta_t - \epsilon^2,$$

where the first equality used $c - \ell = \delta$ which is true by definition. Since we terminate at the first T such that $\delta_T \leq 2\sqrt{3}\epsilon$, we require at most $O(1/\epsilon^2)$ iterations. By Lemma 3, each iteration takes $O(\log(1/\epsilon))$ oracle calls. □

C Generalization analysis

To bound the generalization error of Algorithm 1, we start by defining the central random variable in the analysis. At round i, recall our notation $Q_i(y) = \mathbb{1}$ (query y on example x_i) which indicates the query rule. The central random variable we study is,

$$M_i(g; y) \quad \triangleq \left((g(x_i) - c_i(y))^2 - (f^\star(x_i; y) - c_i(y))^2\right) Q_i(y). \tag{12}$$

Here (x_i, c_i) is the ith example and cost presented to the algorithm. For simplicity, we write M_i when the dependence on g and y is clear from context. For a vector regressor f, we write

$$M_i(f; y) \triangleq M_i(f(\cdot; y); y).$$

We also recall some of the key constants and notations which were defined in Algorithm 1 and are heavily used throughout this appendix.

$$\Delta_i = \frac{\kappa\epsilon_{i-1}}{i-1}, \quad \epsilon_i = \left(\frac{n}{i}\right)^\beta \log\left(\frac{2n^2|\mathcal{G}|K}{\delta}\right), \quad \kappa = 80.$$

$$\widehat{R}_i(g; y) = \frac{1}{i-1}\sum_{j=1}^{i-1}\left[(g(x_j) - c_j(y))^2 Q_j(y)\right].$$

$$g_{i,y} = \operatorname*{argmin}_{g\in\mathcal{G}} \widehat{R}_i(g; y), \quad \text{and} \quad f_i = \{g_{i,y}\}_{y=1}^K.$$

$$\mathcal{G}_i(y) = \{g \in \mathcal{G} | \forall y,\ \widehat{R}_i(g; y) \leq \widehat{R}_i(g_{i,y}; y) + \Delta_i\},$$

$$\mathcal{F}_i = \{f \in \mathcal{G}^K | \forall y,\ \widehat{R}_i(f(\cdot; y); y) \leq \widehat{R}_i(g_{i,y}; y) + \Delta_i\}.$$

For Δ_1 we use the convention that $1/0 = \infty$ so the initial radius is infinite. Let $\mathbb{E}_i[\cdot]$ and $\mathrm{Var}_i[\cdot]$ denote the expectation and variance conditioned on all randomness up to round $i - 1$. With these definitions, we turn to several supporting claims.

C.1 Supporting Lemmata

Theorem 7 (Freedman-type Inequality [8, 2]). *Let $X_1, \ldots, X_T$ be a sequence of real-valued random variables. Assume for all $t \in \{1, \ldots, T\}$ that $|X_t| \leq R$ and $\mathbb{E}[X_t|X_1, \ldots, X_{t-1}] = 0$. Define $S = \sum_{t=1}^T X_t$ and $V = \sum_{t=1}^T \mathbb{E}[X_t^2|X_1, \ldots, X_{t-1}]$. For any $\delta \in (0, 1)$ and $\lambda \in [0, 1/R]$, with probability at least $1 - \delta$,*

$$S \leq (e - 2)\lambda V + \frac{\ln(1/\delta)}{\lambda}$$

Lemma 5 (Concentration of squared loss)**.** *For any $\delta \in (0,1)$, with probability at least $1-\delta$, the following holds for all $g \in \mathcal{G}, y \in Y, i \in [n], t \in [n]$,*

$$\left|\sum_{j=i}^{i+t-1} \mathbb{E}_j[M_j] - \sum_{j=i}^{i+t-1} M_j\right| \leq 2\sqrt{\sum_{j=i}^{i+t-1} \underset{j}{\mathrm{Var}}[M_j]\nu_n} + 2\nu_n$$

where $\nu_n \triangleq \log\left(\frac{2n^2|\mathcal{G}|K}{\delta}\right)$.

Note that $\epsilon_t = \frac{n}{t}^\beta \nu_n$, is a scaled version of the confidence bound here, where the scaling shrinks polynomially with t.

Proof. First observe that by the rescaling of the failure parameter, we can apply Freedman's inequality for each i, t, y, g and for each tail and a union bound proves the result.

We now apply the Freedman-type inequality in Theorem 7. For a fixed $g \in \mathcal{G}, y \in Y$, the random variable M_i is measurable with respect to the filtration $\{(x_j, \{(c(x_j; y), Q_j(y))\}_{y\in Y})\}_{j=1}^{i})$, so $M_i - \mathbb{E}_i[M_i]$ forms a martingale difference sequence, where $\mathbb{E}_i[\cdot]$ denotes expectation conditioned on all randomness up to round $i-1$. Moreover $M_i - \mathbb{E}_i[M_i]$ and $\mathbb{E}_i[M_i] - M_i$ are both conditionally centered and clearly at most 2. Thus Freedman's inequality gives,

$$\sum_{j=i}^{i+t-1} M_j - \mathbb{E}_j[M_j] \leq 2\sqrt{\sum_{j=i}^{i+t-1} \underset{j}{\mathrm{Var}}[M_j]\nu_n} + 2\nu_n,$$

except with probability $\frac{\delta}{2n^2|\mathcal{G}|K}$. This follows by observing that $(e-2) \leq 1$ and setting $\lambda = \sqrt{\nu_n/V}$, provided it meets the constraint $\lambda \leq 1/R$. Otherwise we set $\lambda = 1/R$ and use the fact that $1/R \leq \sqrt{\nu_n/V}$.

The bound on the right hand side also holds for the lower tail, again except with same probability. Thus a union bound over both tails, all $g \in \mathcal{G}, y \in Y$ and pairs i, t gives the result. □

Lemma 6 (Bounding variance of regression regret)**.** *We have for all $(g, y) \in \mathcal{G} \times Y$,*

$$\begin{aligned}\mathbb{E}_i[M_i] &= \mathbb{E}_i\left[Q_i(y)(g(x_i) - f^\star(x_i; y))^2\right], \\ \underset{i}{\mathrm{Var}}[M_i] &\leq 4\mathbb{E}_i[M_i].\end{aligned}$$

Proof. We take expectation of M_i over the cost conditioned on a fixed example $x_i = x$ and a fixed query outcome $Q_i(y)$:

$$\begin{aligned}\mathbb{E}[M_i \mid x_i = x, Q_i(y)] &= Q_i(y) \times \mathbb{E}_c[g(x)^2 - f^\star(x;y)^2 - 2c(y)(g(x) - f^\star(x;y)) \mid x_i = x] \\ &= Q_i(y)\big(g(x)^2 - f^\star(x;y)^2 - 2f^\star(x;y)(g(x) - f^\star(x;y))\big) \\ &= Q_i(y)(g(x) - f^\star(x;y))^2.\end{aligned}$$

The second equality is by Assumption 1, which implies $\mathbb{E}[c(y) \mid x_i = x] = f^\star(x; y)$. Taking expectation over x_i and $Q_i(y)$, we have

$$\mathbb{E}_i[M_i] \;=\; \mathbb{E}_i\left[Q_i(y)(g(x_i) - f^\star(x_i; y))^2\right].$$

For the variance:

$$\begin{aligned}
\operatorname*{Var}_i[M_i] &\leq \mathbb{E}_i[M_i^2] \\
&= \cdot\mathbb{E}_i\left[Q_i(y)(g(x_i) - f^\star(x_i;y))^2(g(x_i) + f^\star(x_i;y) - 2c(y))^2\right] \\
&\leq 4 \cdot \mathbb{E}_i\left[Q_i(y)(g(x_i) - f^\star(x_i;y))^2\right] \\
&= 4\mathbb{E}_i[M_i].
\end{aligned}$$

□

Lemma 7 (Sharp cost-sensitive bound)**.** *For all $i > 0$ if $f^\star \in \mathcal{F}_i$, then for all $f \in \mathcal{F}_i$*

$$\mathbb{E}_{x,c}[c(h_f(x)) - c(h_{f^\star}(x))] \leq \min_{\zeta > 0}\left\{\zeta P_\zeta + \mathbb{1}\left(\zeta \leq 2\eta_i\right)2\eta_i + \frac{4\eta_i^2}{\zeta} + \frac{6}{\zeta}\sum_y \mathbb{E}_i\left[M_i(f;y)\right]\right\},$$

where $P_\zeta = \Pr_{x\sim\mathcal{D}}[\min_{y \neq h_{f^\star}(x)} f^\star(x,y) \leq f^\star(x, h_{f^\star}(x)) + \zeta]$ is the probability that the expected cost of second best and best label are within ζ of each other.

Proof. Let $y(x) = h_f(x)$ and $y^\star(x) = h_{f^\star}(x)$ for shorthand. Define $S_\zeta(x) = \mathbb{1}\left(f^\star(x, y(x)) \leq f^\star(x, y^\star(x)) + \zeta\right)$ and $S'_\zeta(x) = \mathbb{1}\left(\min_{y \neq y^\star(x)} f^\star(x,y) \leq f^\star(x, y^\star(x)) + \zeta\right)$. Observe that for fixed ζ, $S_\zeta(x)\mathbb{1}\left(y(x) \neq y^\star(x)\right) \leq S'_\zeta(x)$ for all x. We can also majorize the complementary indicator to obtain the inequality

$$S_\zeta^C(x) \leq \frac{(f^\star(x, y(x)) - f^\star(x, y^\star(x))}{\zeta}.$$

We begin with the definition of realizability, which gives

$$\begin{aligned}
\mathbb{E}_{x,c}[c(h_f(x)) - c(h_{f^\star}(x)] &= \mathbb{E}_x\left[f^\star(x, y(x)) - f^\star(x, y^\star(x))\mathbb{1}\left(y(x) \neq y^\star(x)\right)\right] \\
&= \mathbb{E}_x\left[\left(S_\zeta(x) + S_\zeta^C(x)\right)\left(f^\star(x, y(x)) - f^\star(x, y^\star(x))\right)\mathbb{1}\left(y(x) \neq y^\star(x)\right)\right] \\
&\leq \zeta\mathbb{E}_x S'_\zeta(x) + \mathbb{E}_i\left[S_\zeta^C(x)\mathbb{1}\left(y(x) \neq y^\star(x)\right)\left(f^\star(x, y(x)) - f^\star(x, y^\star(x))\right)\right]
\end{aligned}$$

The first term here is exactly the ζP_ζ term in the bound. We now focus on the second term, which depends on our query rule. For this we must consider three cases.

Case 1. If both $y(x)$ and $y^\star(x)$ are not queried, then it must be the case that both have small cost ranges. This follows since $f \in \mathcal{F}_i$ and $h_f(x) = y(x)$ so $y^\star(x)$ does not dominate $y(x)$. Moreover, since the cost ranges are small on both $y(x)$ and $y^\star(x)$, since we know that $f^\star$ is well separated under event $S_\zeta^C(x)$, the relationship between ζ and η_i governs whether we make a mistake or not. Specifically, we get that $S_\zeta^C(x)\mathbb{1}\left(y(x) \neq y^\star(x)\right)\mathbb{1}\left(\text{no query}\right) \leq \mathbb{1}\left(\zeta \leq 2\eta_i\right)$ at round i. In other words, if we do not query and the separation is big but we make a mistake, then it must mean that the cost range threshold η_i is also big.

Using this argument, we can bound the second term as,

$$\begin{aligned}
&\mathbb{E}_i\left[S_\zeta^C(x)\mathbb{1}\left(y(x) \neq y^\star(x)\right)\mathbb{1}\left(y(x), y^\star(x) \text{ not queried}\right)\left(f^\star(x, y(x)) - f^\star(x, y^\star(x))\right)\right] \\
&\leq \mathbb{E}_i\left[S_\zeta^C(x)\mathbb{1}\left(y(x) \neq y^\star(x)\right)\mathbb{1}\left(y(x), y^\star(x) \text{ not queried}\right)\left(f^\star(x, y(x)) - f(x, y(x) + f(x, y^\star(x)) - f^\star(x, y^\star(x))\right)\right] \\
&\leq \mathbb{E}_i\left[S_\zeta^C(x)\mathbb{1}\left(y(x) \neq y^\star(x)\right)\mathbb{1}\left(y(x), y^\star(x) \text{ not queried}\right)2\eta_i\right] \\
&\leq \mathbb{E}_i\left[\mathbb{1}\left(\zeta \leq 2\eta_i\right)2\gamma\right] = \mathbb{1}\left(\zeta \leq 2\eta_i\right)2\eta_i.
\end{aligned}$$

Case 2. If both $y(x)$ and $y^\star(x)$ are queried, we can easily relate the second term to the square loss,

$$\begin{aligned}
&\mathbb{E}_i\left[S^C_\zeta(x)\mathbb{1}\left(y(x), y^\star(x) \text{ both queried}\right)\left(f^\star(x,y(x)) - f^\star(x,y^\star(x))\right)\right]\\
&\le \frac{1}{\zeta}\mathbb{E}_i\left[\mathbb{1}\left(y(x), y^\star(x) \text{ both queried}\right)\left(f^\star(x,y(x)) - f^\star(x,y^\star(x))\right)^2\right]\\
&\le \frac{1}{\zeta}\mathbb{E}_i\left[\mathbb{1}\left(y(x), y^\star(x) \text{ both queried}\right)\left(f^\star(x,y(x)) - f(x,y(x)) + f(x,y^\star(x)) - f^\star(x,y^\star(x))\right)^2\right]\\
&\le \frac{2}{\zeta}\mathbb{E}_i\left[\mathbb{1}\left(y(x) \text{ queried}\right)\left(f^\star(x,y(x)) - f(x,y(x))\right)^2 + \mathbb{1}\left(y^\star(x) \text{ queried}\right)\left(f(x,y^\star(x)) - f^\star(x,y^\star(x))\right)^2\right]\\
&\le \frac{2}{\zeta}\sum_y \mathbb{E}_i\left[Q_i(y)(f^\star(x,y(x)) - f(x,y(x)))^2\right] = \frac{2}{\zeta}\sum_y \mathbb{E}_i\left[M_i(f;y)\right].
\end{aligned}$$

Passing from the second to third line here is justified by the fact that $f^\star(x,y(x)) \ge f^\star(x,y^\star(x))$ and $f(x,y(x)) \le f(x,y^\star(x))$ so we added two non-negative quantities together. The last step uses Lemma 6. While not written, we also use the event $\mathbb{1}\left(y(x) \ne y^\star(x)\right)$ to avoid losing a factor of 2.

Case 3. The last case is if one label is queried and the other is not. Both cases here are analogous, so we do the derivation for when $y(x)$ is queried but $y^\star(x)$ is not. Since in this case, $y^\star(x)$ is not dominated ($h_f(x)$ is never dominated provided $f \in \mathcal{F}_i$), we know that the cost range for $y^\star(x)$ must be small. Using this fact, and essentially the same argument as in case 2, we get

$$\begin{aligned}
&\mathbb{E}_i\left[S^C_\zeta(x)\mathbb{1}\left(y(x) \text{ queried}, y^\star(x) \text{ not}\right)\left(f^\star(x,y(x)) - f^\star(x,y^\star(x))\right)\right]\\
&\frac{1}{\zeta}\mathbb{E}_i\left[\mathbb{1}\left(y(x) \text{ queried}, y^\star(x) \text{ not}\right)\left(f^\star(x,y(x)) - f^\star(x,y^\star(x))\right)^2\right]\\
&\le \frac{2}{\zeta}\mathbb{E}_i\left[\mathbb{1}\left(y(x) \text{ queried}, y^\star(x) \text{ not}\right)\left(f^\star(x,y(x)) - f(x,y(x))\right)^2 + \left(f(x,y^\star(x)) - f^\star(x,y^\star(x))\right)^2\right]\\
&\le \frac{2\eta_i^2}{\zeta} + \frac{2}{\zeta}\mathbb{E}_i\left[\mathbb{1}\left(y(x) \text{ queried}\right)\left(f^\star(x,y(x)) - f(x,y(x))\right)^2\right]\\
&\le \frac{2\eta_i^2}{\zeta} + \frac{2}{\zeta}\sum_y \mathbb{E}_i\left[M_i(f;y)\right].
\end{aligned}$$

We also obtain this term for the other case where $y^\star(x)$ is queried by $y(x)$ is not.

To summarize, adding up the contributions from these cases (which is an over-estimate since at most one case can occur and all are non-negative), we get

$$\mathbb{E}_{x,c}[c(h_f(x)) - c(h_{f^\star}(x)] \le \zeta P_\zeta + \mathbb{1}\left(\zeta \le 2\eta_i\right)2\eta_i + \frac{4\eta_i^2}{\zeta} + \frac{6}{\zeta}\sum_y \mathbb{E}_i\left[M_i(f;y)\right].$$

This bound holds for any ζ, so it holds for the minimum. □

C.2 Proof of Theorem 3

Conditioning on the high-probability event in Lemma 5, we prove the theorem by induction. Define

$$\Delta'_i = \min\{1, \frac{\kappa\nu_n}{i-1}\}, \quad \nu_n = \log\left(\frac{2n^2|\mathcal{G}|K}{\delta}\right).$$

Here and throughout we will make use of the following simple fact, which applies since $i \le n$, so the premultiplier on ϵ_i is at least 1.

Fact 1. *For all $i \in [n]$, we have*

$$\nu_n \le \epsilon_i.$$

Concretely we consider the inductive hypothesis:

$$\forall i \ge 1, \quad \widehat{R}_i(f^\star(\cdot;y);y) \le \min_{g\in\mathcal{G}} \widehat{R}_i(g;y) + \frac{c_0\nu_n}{i-1} \quad \text{and} \quad \mathbb{E}_{x,c}[c(h_{f_i}(x)) - c(h_{f^\star}(x))] \le \min_{\zeta>0}\left\{\zeta P_\zeta + \frac{2K\Delta_i'}{\zeta}\right\} \tag{13}$$

where $c_0 = 10$. The first claim in particular implies that $f^\star(\cdot;y) \in \mathcal{F}_i$ since we chose $\Delta_{i+1} = \kappa\epsilon_i/i$ and using Fact 1. For the base case $i = 1$, observe that the right hand side of the first inequality is infinity but the empirical squared loss is 0 for all regressors. Hence the first claim is trivially satisfied. Moreover, because the excess cost-sensitive classification risk is always upper-bounded by 1, it is trivially bounded by $\frac{2K\Delta_1'}{\zeta}$ for any $\zeta \in [0,1]$. For $\zeta > 1$, we have $\zeta P_\zeta = \zeta$ so again the bound is trivial.

Now assume the inductive hypothesis holds for the first i rounds, $i \ge 1$. We want to analyze the set $\mathcal{F}_{i+1}$, which is computed at the end of the i^{th} iteration of Algorithm 1 based on i examples (technically the beginning of the $(i+1)^{\text{st}}$ iteration). Invoking Lemma 5, with parameters 1 and i, and Lemma 6, we have for all $(g,y) \in \mathcal{G} \times Y$,

$$\begin{aligned}
\sum_{j=1}^{i} \mathbb{E}_j[M_j(g;y)] - \sum_{j=1}^{i} M_j(g;y) &\le 2\sqrt{4\nu_n \sum_{j=1}^{i} \mathbb{E}_j[M_j(g;y)] + 2\nu_n} \\
&\le 2\left(4\nu_n + \frac{1}{4}\sum_{j=1}^{i} \mathbb{E}_j[M_j(g;y)]\right) + 2\nu_n \\
&= 10\nu_n + \frac{1}{2}\sum_{j=1}^{i} \mathbb{E}_j[M_j].
\end{aligned}$$

This bound implies that

$$-\sum_{j=1}^{i} M_j \;\le\; 10\nu_n, \quad \text{(since } \mathbb{E}_j[M_j(g;y)] \ge 0 \text{ by Lemma 6)}$$

and therefore

$$\widehat{R}_{i+1}(f^\star(\cdot;y);y) \;\le\; \widehat{R}_{i+1}(g;y) + \frac{c_0\nu_n}{i}. \tag{14}$$

Since this bound applies for all $g \in \mathcal{G}$, it proves the first part of the inductive claim.

Next we prove that the empirical squared loss minimizer f_{i+1} after iteration i has small excess risk. Fix some label y. To simplify notations, we drop the dependence on y and define for any j:

$$g_j \triangleq f_j(\cdot;y), \quad g^\star \triangleq f^\star(\cdot;y), \quad \mathcal{G}_j \triangleq \mathcal{G}_j(y), \quad \widehat{R}_j(g) \triangleq \widehat{R}_j(g;y).$$

Let M_j be defined for g_{i+1} and y according to Eq. (12). We first prove that since g_{i+1} is the empirical loss minimizer at round i for label y, it must have been in the version space $\mathcal{G}_j(y)$ for all $j \in \{1,\ldots,i+1\}$.

Because g_{i+1} is the loss minimizer for label y after round i, we have

$$\sum_{j=1}^{i} M_j = \sum_{j=1}^{i} M_j(g_{i+1}; y) \ \le \ 0.$$

Now suppose $g_{i+1} \notin \mathcal{G}_{t+1}$ for some $t \in \{0, \dots, i\}$. We have

$$\begin{aligned} \sum_{j=1}^{t} M_j &= t\left(\widehat{R}_{t+1}(g_{i+1}) - \widehat{R}_{t+1}(g^\star)\right) \\ &= t\left(\widehat{R}_{t+1}(g_{i+1}) - \widehat{R}_{t+1}(g_{t+1}) + \widehat{R}_{t+1}(g_{t+1}) - \widehat{R}_{t+1}(g^\star)\right) \\ &\ge \kappa\epsilon_t - c_0\nu_n. \end{aligned} \tag{15}$$

The last inequality here follows since $g_{i+1} \notin \mathcal{G}_{t+1}$ so it must have $\widehat{R}_{t+1}(g_{i+1}) - \widehat{R}_{t+1}(g_{t+1}) \ge \kappa\epsilon_t/t$ by the elimination rule. Simultaneously, we use Eq. (14) which lower bounds the second term. Combining this inequality with the fact that $\sum_{j=1}^{i} M_j \le 0$ gives

$$\sum_{j=t+1}^{i} M_j \le c_0\nu_n - \kappa\epsilon_t. \tag{16}$$

Applying Lemmas 5 and 6 along with the inequality $\sqrt{4ab} \le a/\alpha + \alpha b$ for all $\alpha > 0$, gives

$$\sum_{j=t+1}^{i} \mathbb{E}_j[M_j] - \sum_{j=t+1}^{i} M_j \ \le \ 2\sqrt{4\nu_n \sum_{j=t+1}^{i} \mathbb{E}_j[M_j]} + 2\nu_n \le \frac{1}{2}\sum_{j=t+1}^{i} \mathbb{E}_j[M_j] + 10\nu_n. \tag{17}$$

Combining the last inequality and Eq. (16), we get

$$\sum_{j=t+1}^{i} \mathbb{E}_j[M_j] \le 20\nu_n + 2c_0\nu_n - 2\kappa\epsilon_t \le (40 - 2\kappa)\epsilon_t < 0.$$

The strict inequality here is based on Fact 1 and the parameter setting $\kappa = 80$. This is a contradiction since $\mathbb{E}_j[M_j]$ is a quadratic form and hence non-negative by Lemma 6. The same analysis applies to every y. Therefore, we know that the empirical square loss vector regressor f_{i+1} is in $\mathcal{F}_j$ for all $j \in \{1, \dots, i+1\}$, and hence we can apply Lemma 7 for all of these rounds, to obtain

$$\begin{aligned} &i\big(\mathbb{E}_{x,c}[c(h_{f_{i+1}}(x)) - c(h_{f^\star}(x))]\big) \\ &\le \min_{\zeta>0}\left\{\sum_{j=1}^{i}\left(\zeta P_\zeta + \mathbb{1}\left(\zeta \le 2\eta_j\right)2\eta_j + \frac{4\eta_j^2}{\zeta} + \frac{6}{\zeta}\sum_y \mathbb{E}_j\left[M_j(f_{i+1}; y)\right]\right)\right\} \\ &\le \min_{\zeta>0}\left\{i\zeta P_\zeta + \sum_{j=1}^{i}\left(\mathbb{1}\left(\zeta \le 2\eta_j\right)2\eta_j + \frac{4\eta_j^2}{\zeta} + \frac{6}{\zeta}\sum_y \mathbb{E}_j\left[M_j(f_{i+1}; y)\right]\right)\right\}. \end{aligned}$$

We study the four terms separately. The first one is straightforward and contributes ζP_ζ to the instantaneous cost sensitive regret. Using our definition of $\eta_j = 1/\sqrt{j}$ the second term can be bounded as

$$\sum_{j=1}^{i} \mathbb{1}\left(\zeta < 2\eta_j\right) 2\eta_j = \sum_{j=1}^{\lceil 4/\zeta^2 \rceil} \frac{2}{\sqrt{j}} \leq 4\sqrt{\lceil 4/\zeta^2 \rceil} \leq \frac{12}{\zeta}.$$

The inequality above, $\sum_{i=1}^{n} \frac{1}{\sqrt{i}} \leq 2\sqrt{n}$, is well known. For the third term, using our definition of η_j gives

$$\sum_{j=1}^{i} \frac{4\eta_j^2}{\zeta} = \frac{4}{\zeta}\sum_{j=1}^{i}\frac{1}{j} \leq \frac{4}{\zeta}(1+\log(i)).$$

Finally, the fourth term can be bounded using Lemma 5 (Eq. (17) with $t = 0$), which reveals

$$\sum_{j=1}^{i} \mathbb{E}_j[M_j] \leq 2\sum_{j=1}^{i} M_j + 20\nu_n$$

Since for each y, $\sum_{j=1}^{i} M_j(f_i; y) \leq 0$ for the empirical square loss minimizer (which is what we are considering now), we get

$$\frac{6}{\zeta}\sum_{y}\sum_{j=1}^{i}\mathbb{E}_j[M_j(f_{i+1}; y)] \leq \frac{120}{\zeta}K\nu_n.$$

And hence, we obtain

$$\begin{aligned}\mathbb{E}_{x,c}[c(x; h_{f_{i+1}}(x)) - c(x; h_{f^\star}(x))] &\leq \min_{\zeta > 0}\left\{\zeta P_\zeta + \frac{1}{\zeta i}\left(4\log(i) + 16 + 120K\nu_n\right)\right\}\\ &\leq \min_{\zeta>0}\left\{\zeta P_\zeta + \frac{140K\nu_n}{\zeta i}\right\} \leq \min_{\zeta>0}\left\{\zeta P_\zeta + \frac{2\kappa K\nu_n}{\zeta i}\right\}\end{aligned}$$

To obtain this last bound, we observe that $1 \leq \log(i) \leq \nu_n$ under our assumption that $\delta < 1/e$ so the coefficient in the numerator is at most 140. The inductive claim follows by the definition of Δ'_{i+1}. Or more precisely, if $\Delta'_{i+1} = 1$ then the inductive claim is trivial and otherwise we have proved what is required.

D Label complexity analysis

D.1 Supporting Lemmata

Our label complexity analysis builds on the following lemma, which uses the sets $\mathcal{G}_i^\star$ and $\mathcal{G}_i$ whose definitions we reproduce here.

$$\mathcal{G}_i(\Delta; y) \triangleq \{g \mid \widehat{R}_i(g; y) - \min_{g' \in \mathcal{G}} \widehat{R}_i(g'; y) \leq \Delta\}, \tag{18}$$

$$\mathcal{G}_i^\star(\Delta; y) \triangleq \left\{g \;\middle|\; \frac{1}{i}\sum_{j=1}^{i} Q_j(y)(g(x_j) - f^\star(x_j; y))^2 \leq \Delta\right\}. \tag{19}$$

Throughout we use the definitions.

$$\Delta_i \triangleq \kappa\epsilon_{i-1}/(i-1), \kappa \triangleq 80, c_0 \triangleq 10, c_1 \triangleq 25/3, c_2 \triangleq 1/3, \eta_i \triangleq 1/\sqrt{i}.$$
$$\nu_n \triangleq \log\left(\frac{2n^2|\mathcal{G}|K}{\delta}\right)$$

These are the constants defined in Algorithm 1 with some additional numerical constants that we use in the analysis. We also require a new definition:

$$I_\beta(i) = \max\{t \in \mathbb{N} | (t-1) \le (c_2/c_1)^{1/\beta}(i-1)\}. \tag{20}$$

Note that $I_\beta(i)$ is well defined for $i \ge 1$ since the right hand side is non-negative. However $I_\beta(i)$ could be as small as 1. We first study the I_β functional.

Fact 2. *Define $i_\beta \triangleq 2(c_1/c_2)^{1/\beta} + 1$. Then for $i \ge i_\beta$, we have*

$$I_\beta(i) - 1 \ge \max\{(c_2/c_1)^{1/\beta}(i-1)/2, 2\}.$$

Proof. The proof is by direct calculation.

$$I_\beta(i) - 1 = \lfloor (c_2/c_1)^{1/\beta}(i-1) \rfloor \ge \lfloor (c_2/c_1)^{1/\beta}(i_\beta - 1) \rfloor = 2$$
$$I_\beta(i) - 1 \ge (c_2/c_1)^{1/\beta}(i-1) - 1 = (c_2/c_1)^{1/\beta}(i-1) - \frac{(c_2/c_1)^{1/\beta}(i_\beta-1)}{2} \ge \frac{(c_2/c_1)^{1/\beta}(i-1)}{2}.$$

□

We now turn to the more intricate lemmas.

Lemma 8. *For any $\delta \in (0,1)$, with probability at least $1-\delta$, for all $i \ge 1$ and all y,*

$$\mathcal{G}_i^\star(c_2\Delta_i; y) \subset \mathcal{G}_i(\Delta_i; y) \subset \mathcal{G}_i(4\Delta_i; y) \subset \mathcal{G}_i^\star(c_1\Delta_i; y) \subset \mathcal{G}^\star_{I_\beta(i)}(c_2\Delta_{I_\beta(i)}; y),$$

where $I_\beta(i)$ is in Eq. (20).

Proof. The second containment is trivial.

Recall our earlier definition that for a fixed $g \in \mathcal{G}$ and $y \in Y$,

$$M_j \triangleq \left((g(x_j) - c(x_j;y))^2 - (f^\star(x_j;y) - c(x_j;y))^2\right) Q_j(y).$$

Let $\mathbb{E}_c[M_j]$ and $\mathrm{Var}_c[M_j]$ denote the expectation and variance taken with respect to the cost c at round j, conditioned on all randomness up to round $j-1$ and on x_j. Following the same proof for Lemma 6, we have that

$$\mathbb{E}_c[M_j] = Q_j(y)(g(x_j) - f^\star(x_j;y))^2,$$
$$\underset{c}{\mathrm{Var}}[M_j] \le 4\mathbb{E}_c[M_j(g;y)].$$

It is also easy to prove a concentration result similar to Lemma 5 where $\mathbb{E}_j[M_j]$ and $\mathrm{Var}_j[M_j]$ are replaced by $\mathbb{E}_c[M_j]$ and $\mathrm{Var}_c[M_j]$, respectively. Thus we have for any $\delta \in (0,1)$, with probability at least $1-\delta$, the following holds for all $(g,y) \in \mathcal{G} \times Y$ and all $i, t \in [n]$:

$$\left|\sum_{j=i}^{i+t-1} \mathbb{E}_c[M_j] - \sum_{j=i}^{i+t-1} M_j\right| \le 2\sqrt{4\nu_n \sum_{j=i}^{i+t-1} \mathbb{E}_c[M_j]} + 2\nu_n, \tag{21}$$

where $\nu_n = \log\left(\frac{2n^2|\mathcal{G}|K}{\delta}\right)$ as in Lemma 5. This bound, via the inequality $\sqrt{4ab} \le \alpha a + b/\alpha$ implies

$$\sum_{j=i}^{i+t-1} \mathbb{E}_c[M_j] \le 2 \sum_{j=i}^{i+t-1} M_j + 20\nu_n \tag{22}$$

$$\sum_{j=i}^{i+t-1} M_j \le \frac{3}{2} \sum_{j=i}^{i+t-1} \mathbb{E}_c[M_j] + 10\nu_n \tag{23}$$

We start with proving the first containment. Fix some round i, some label y and some $g \in \mathcal{G}_i^\star(c_2\Delta_i; y)$. Conditioning on the above high-probability event, and starting with Eq. (23) we have

$$\begin{aligned} \sum_{j=1}^{i-1} M_j &\le \frac{3}{2} \cdot \left(\sum_{j=1}^{i-1} \mathbb{E}_c[M_j]\right) + 10\nu_n \le \frac{3}{2} \cdot (i-1) \cdot c_2\Delta_i + 10\nu_n \\ &= \frac{3}{2} c_2 \kappa \epsilon_{i-1} + 10\nu_n \le \left(\frac{\kappa}{2} + c_0\right) \epsilon_{i-1}. \end{aligned}$$

Above, the second inequality is by

$$\sum_{j=1}^{i-1} \mathbb{E}_c[M_j] \;=\; \sum_{j=1}^{i-1} Q_j(y)(g(x_j) - f^\star(x_j; y))^2 \;\le\; c_2\Delta_i \times (i-1)$$

since $g \in \mathcal{G}_i^\star(c_2\Delta_i; y)$, and the final inequality uses $\nu_n \le \epsilon_{i-1}$ (Fact 1) and our choices of κ, c_0 and c_2. Using the above bound and with $g_i = \operatorname{argmin}_{g \in \mathcal{G}} \widehat{R}_i(g; y)$, we have

$$\begin{aligned} (i-1) \cdot \left(\widehat{R}_i(g; y) - \widehat{R}_i(g_i; y)\right) &= \sum_{j=1}^{i-1} M_j + (i-1)\left(\widehat{R}_i(f^\star; y) - \widehat{R}_i(g_i; y)\right) \\ &\le (\kappa/2 + c_0)\epsilon_{i-1} + c_0\nu_n \le \kappa\epsilon_{i-1}, \end{aligned}$$

where the first inequality is by the above upper bound on $\sum_{j=1}^{i-1} M_j$ and Eq. (14), which upper bounds the excess empirical square loss of $f^\star$. Thus, $g \in \mathcal{G}_i(\Delta_i; y) \subset \mathcal{G}_i(4\Delta_i; y)$.

To prove the third containment, we fix some i, y, and $g \in \mathcal{G}_i(4\Delta_i; y)$. Starting from (22) we have

$$\begin{aligned} \sum_{j=1}^{i-1} \mathbb{E}_c[M_j] &\le 2 \sum_{j=1}^{i-1} M_j + 20\nu_n \\ &= 2(i-1) \cdot (\widehat{R}_i(g; y) - \widehat{R}_i(f^\star; y)) + 20\nu_n \\ &\le 2(i-1) \cdot (\widehat{R}_i(g; y) - \widehat{R}_i(g_i; y)) + 20\nu_n \\ &\le 8\kappa\epsilon_{i-1} + 20\nu_n \\ &\le c_1\kappa\epsilon_{i-1}, \end{aligned}$$

where the second inequality is by the fact that g_i is the minimizer of the squared loss at round i for label y, the third inequality is by $g \in \mathcal{G}_i(4\Delta_i; y)$, and the last inequality is by $\nu_n \le \epsilon_{i-1}$ (Fact 1) and our choices of c_1 and κ. Thus, $g \in \mathcal{G}_i^\star(c_1\Delta_i; y)$.

For the final containment, observe that

$$(i-1)c_1\Delta_i = c_1\kappa\epsilon_{i-1} = c_1\kappa\left(\left(\frac{n}{i-1}\right)^{\beta}\nu_n\right) = c_2\kappa\left(\left[\left(\frac{c_1}{c_2}\right)^{1/\beta}\frac{n}{i-1}\right]^{\beta}\nu_n\right)$$

Using the definition of $I_\beta(i)$ in Eq. (20), we get that $(i-1)c_1\Delta_i \le (I_\beta(i)-1)c_2\Delta_{I_\beta(i)}$. Of course we always have $I_\beta(i)-1 \le i-1$ since $c_2 \le c_1$. Hence,

$$\sum_{j=1}^{I_\beta(i)-1}\mathbb{E}_j Q_j(y)(g(x_j)-f^\star(x_j;y))^2 \le \sum_{j=1}^{i-1}\mathbb{E}_j Q_j(y)(g(x_j)-f^\star(x_j;y))^2 \le (i-1)c_1\Delta_i \le (I_\beta(i)-1)c_2\Delta_{I_\beta(i)}.$$

Thus we get that $\mathcal{G}_i^\star(c_1\Delta_i) \subset \mathcal{G}^\star_{I_\beta(i)}(c_2\Delta_{I_\beta(i)})$. □

Before bounding the label complexity, we first prove the following regret bound:

Lemma 9. *For any $\delta \le 1/e$, with probability at least $1-\delta$, for all $i \ge 1$ and for all vector regressors $f \in \mathcal{F}_i^\star(c_2\Delta_i) \triangleq \prod_y \mathcal{G}_i^\star(c_2\Delta_i;y)$,*

$$\mathbb{E}_{x,c}\left[c(x,h_f(x)) - c(x,h_{f^\star}(x))\right] \le \min_{\zeta>0}\left\{\zeta P_\zeta + \frac{14K\Delta_i}{\zeta}\right\}.$$

Note that this cost-sensitive regret bound is polynomially worse than the one in Theorem 3 that we prove just for the empirical risk minimizer f_i. This is because we set the confidence radius Δ_i using a polynomial function of n/i, which will be important for our label complexity analysis.

Proof. The proof follows a similar argument to that of Lemma 7 in that we must argue that each $g \in \mathcal{G}_i^\star(c_2\Delta_i;y)$ is involved in driving the query rule for a large fraction of the rounds. First observe that $f^\star \in \mathcal{F}_i^\star(c_2\Delta_i)$ for $i \ge 1$ by the definition of $\mathcal{F}_i^\star$.

Next, fix a label y and a function $g \in \mathcal{G}^\star_{i+1}(c_2\Delta_{i+1};y)$ for $i \ge 0$. We prove that $g \in \mathcal{G}_{t+1}(\Delta_t)$ for all $t \in \{0,\ldots,i\}$. In search of a contradiction, suppose that $g \notin \mathcal{G}_{t+1}(\Delta_{t+1})$ for some $t \in \{0,\ldots,i\}$. First, since $g \in \mathcal{G}^\star_{i+1}(c_2\Delta_{i+1};y)$, using the Freedman-style deviation bound in Eq. (23), we have

$$\sum_{j=1}^{i} M_j \le \frac{3}{2}\sum_{j=1}^{i}\mathbb{E}_c[M_j] + 10\nu_n \le \left(\frac{3}{2}c_2\kappa + c_0\right)\epsilon_i.$$

Here we also use the definition of $\Delta_{i+1} = \kappa\epsilon_i/i$, $c_0 = 10$, and Fact 1.

At the same time, since $g \notin \mathcal{G}_{t+1}(\Delta_{t+1};y)$, we know that

$$\Delta_{t+1} < \hat{R}_{t+1}(g) - \hat{R}_{t+1}(g_{t+1}) < \hat{R}_{t+1}(g) - \hat{R}_{t+1}(g^\star) + \frac{c_0\nu_n}{t}.$$

The last inequality uses Eq. (14). Together with the above, this implies that

$$\sum_{j=t+1}^{i} M_j \le \left(\frac{3}{2}c_2\kappa + c_0\right)\epsilon_i - \kappa\epsilon_t + c_0\nu_n.$$

Now, since $i \geq t$ and $\beta \in (0,1)$, we get that $\epsilon_i < \epsilon_t$. Now, using Eq. (22) as before, we get

$$\sum_{j=t+1}^{i} \mathbb{E}_c[M_j] \leq 2 \sum_{j=t+1}^{i} M_j + 20\nu_n \leq 2\left(\frac{3}{2}c_2\kappa + c_0\right)\epsilon_i - 2\kappa\epsilon_t + 4c_0\nu_n \leq (-\kappa + 6c_0)\epsilon_t < 0.$$

The last non-strict inequality follows from the fact that $\epsilon_t \geq \epsilon_i \geq \nu_n$ since $i \geq t$, and then the strict inequality is by our choices for the constants. This is a contradiction since the left hand side is a quadratic form and so, $g \in \mathcal{G}_{t+1}(\Delta_{t+1})$ for all $t \in \{0, \ldots, i\}$.

This argument applies for all y, and hence, for these rounds we may apply Lemma 7, so that for all regressors $f \in \mathcal{F}_i^\star(c_2\Delta_{i+1})$,

$$\begin{aligned} i \cdot (\mathbb{E}_{x,c}[c(x, h_f(x)) - c(x; h_{f^\star}(x))]) &\leq \min_{\zeta > 0} \left\{ i\zeta P_\zeta + \sum_{j=1}^{i} \left(\mathbb{1}\left(\zeta \leq 2\eta_j\right) 2\eta_j + \frac{4\eta_j^2}{\zeta} + \frac{6}{\zeta} \sum_y \mathbb{E}_j\left[M_j(f;y)\right] \right) \right\} \\ &\leq \min_{\zeta > 0} \left\{ i\zeta P_\zeta + \frac{16 + 4\log(i)}{\zeta} + \frac{6}{\zeta} \sum_y \sum_{j=1}^{i} \mathbb{E}_j\left[M_j(f;y)\right] \right\}. \end{aligned}$$

The last inequality here uses identical bounds as the proof of Theorem 3.

In a similar way to (17), we use Lemma 5 to obtain

$$\begin{aligned} \sum_{j=1}^{i} \mathbb{E}_j[M_j(f;y)] \leq 2\sum_{j=1}^{i} M_j(f;y) + 20\nu_n &= 2i \cdot \left(\widehat{R}_{i+1}(f;y) - \widehat{R}_{i+1}(f^\star;y)\right) + 20\nu_n \\ &\leq 2i \cdot \left(\widehat{R}_{i+1}(f;y) - \widehat{R}_{i+1}(f_{i+1};y)\right) + 20\nu_n \\ &\leq (2\kappa + 20)\epsilon_i \end{aligned}$$

The last bound uses the definition of Δ_{i+1} and Fact 1, along with the fact that $\mathcal{G}_{i+1}^\star(c_2\Delta_{i+1}; y) \subset \mathcal{G}_{i+1}(\Delta_{i+1}; y)$ so we know the empirical risk to f_{i+1} is controlled. Finally, we collect the latter three terms and collect the constant $6(2\kappa + 20) + 20$ (which requires $\delta < 1/e$). This gives,

$$\mathbb{E}_{x,c}\left[c(x, h_f(x)) - c(x, h_{f^\star}(x))\right] \leq \min_{\zeta > 0} \left\{ \zeta P_\zeta + \frac{14\kappa K\epsilon_i}{i\zeta} \right\}.$$

This proves the statement since we are considering $f \in \mathcal{F}_{i+1}^\star(c_2\Delta_{i+1})$ and $\kappa\epsilon_i/i = \Delta_{i+1}$. □

For the rest of the analysis, it will be convenient to introduce the shorthand $\widehat{\gamma}(x_i, y) = \widehat{c}_+(x_i, y) - \widehat{c}_-(x_i, y)$, where $\widehat{c}_+(x_i, y)$ and $\widehat{c}_-(x_i, y)$ are the approximate maximum and minimum costs computed in Algorithm 1 at round i.

Lemma 10 (Cost Range Translation). *Fix i and suppose that the conclusions of Lemmas 8 and 9 hold. Then for any x, y pair, we have*

$$\widehat{\gamma}(x_i, y) \leq \gamma(x_i, y, \mathcal{F}_{csr}(r_{I_\beta(i)})) + \eta_i/2,$$

where $r_i = \min_{\zeta > 0}\left\{\zeta P_\zeta + \frac{14K\Delta_i}{\zeta}\right\}$ and $I_\beta(i)$ is in Eq. (20).

Proof. We have

$$\begin{aligned}\widehat{\gamma}(x_i, y) &\le \gamma(x_i, y, \mathcal{G}_i^\star(c_1\Delta_i; y)) + \frac{\eta_i}{2} && \text{(By Theorem 1, setting of } \epsilon \text{ in Algorithm 1 and Lemma 8)}\\ &\le \gamma(x_i, y, \mathcal{F}_{\text{csr}}(r_{I_\beta(i)})) + \frac{\eta_i}{2} && \text{(By Lemmas 8 and 9)}\end{aligned}$$

□

Lemma 11. *Fix i and suppose that the conclusions of Lemmas 8 and 9 hold. Define* $y_i^\star = \operatorname{argmin}_y f^\star(x_i; y), \bar{y}_i = \operatorname{argmin}_y \widehat{c_+}(x_i, \mathcal{G}_i(y)), \tilde{y}_i = \operatorname{argmin}_{y \ne y_i^\star} \widehat{c_-}(x_i, \mathcal{G}_i(y))$*. Then for* $y \ne y_i^\star$*, we have*

$$y \in Y_i \Rightarrow f^\star(x_i; y) - f^\star(x_i; y_i^\star) - \frac{\eta_i}{2} \le (\gamma(x_i, y) + \gamma(x_i, y_i^\star)),$$

and for $y_i^\star$*:*

$$|Y_i| > 1 \;\wedge\; y_i^\star \in Y_i \Rightarrow f^\star(x_i; \tilde{y}_i) - f^\star(x_i; y_i^\star) - \frac{\eta_i}{2} \le (\gamma(x_i, \tilde{y}_i) + \gamma(x_i, y_i^\star)).$$

In both bounds, all the cost ranges are computed using $\mathcal{F}_{csr}(r_{I_\beta(i)})$*.*

Proof. Suppose $y \ne y_i^\star$

$$\begin{aligned} y \in Y_i &\Rightarrow \widehat{c_-}(x_i, \mathcal{G}_i(y)) \le \widehat{c_+}(x_i, \mathcal{G}_i(\bar{y}_i))\\ &\Rightarrow \widehat{c_-}(x_i, \mathcal{G}_i(y)) \le \widehat{c_+}(x_i, \mathcal{G}_i(y_i^\star))\\ &\Rightarrow c_-(x_i, \mathcal{G}_i^\star(c_1\Delta_i; y)) \le c_+(x_i, \mathcal{G}_i^\star(c_1\Delta_i; y_i^\star)) + \frac{\eta_i}{2}\\ &\Rightarrow f^\star(x_i; y) - \gamma(x_i, \mathcal{G}_i^\star(c_1\Delta_i; y)) \le f^\star(x_i; y_i^\star) + \gamma(x_i, \mathcal{G}_i^\star(c_1\Delta_i; y_i^\star))) + \frac{\eta_i}{2}\\ &\Rightarrow f^\star(x_i; y) - f^\star(x_i; y_i^\star) - \frac{\eta_i}{2} \le \gamma(x_i, \mathcal{G}_i^\star(c_1\Delta_i; y)) + \gamma(x_i, \mathcal{G}_i^\star(c_1\Delta_i; y_i^\star))\\ &\Rightarrow f^\star(x_i; y) - f^\star(x_i; y_i^\star) - \frac{\eta_i}{2} \le \big(\gamma(x_i, y, \mathcal{F}_{\text{csr}}(r_{I_\beta(i)})) + \gamma(x_i, y_i^\star, \mathcal{F}_{\text{csr}}(r_{I_\beta(i)}))\big).\end{aligned}$$

For $y_i^\star$ we need to consider two cases. First assume $y_i^\star = \bar{y}_i$. Then

$$\begin{aligned}|Y_i| > 1 \wedge y_i^\star \in Y_i \wedge y_i^\star = \bar{y}_i &\Rightarrow \widehat{c_-}(x_i, \mathcal{G}_i(\tilde{y}_i)) \le \widehat{c_+}(x_i, \mathcal{G}_i(y_i^\star))\\ &\Rightarrow f^\star(x_i, \tilde{y}_i) - f^\star(x_i, y_i^\star) - \frac{\eta_i}{2} \le \gamma(x_i, \tilde{y}_i) + \gamma(x_i, y_i^\star).\end{aligned}$$

This is true since if $|Y_i| > 1$ then it must be the case that $\tilde{y}_i$ is confused, since it has the minimal lower cost estimate. On the other hand if $y_i^\star \ne \bar{y}_i$ then

$$\begin{aligned}|Y_i| > 1 \wedge y_i^\star \in Y_i \wedge y_i^\star \ne \bar{y}_i &\Rightarrow \widehat{c_+}(x_i, \mathcal{G}_i(\bar{y}_i)) \le \widehat{c_+}(x_i, \mathcal{G}_i(y_i^\star))\\ &\Rightarrow \widehat{c_-}(x_i, \mathcal{G}_i(\tilde{y}_i)) \le \widehat{c_+}(x_i, \mathcal{G}_i(y_i^\star))\\ &\Rightarrow f^\star(x_i, \tilde{y}_i) - f^\star(x_i, y_i^\star) - \frac{\eta_i}{2} \le \gamma(x_i, \tilde{y}_i) + \gamma(x_i, y_i^\star).\end{aligned}$$

The second step here is because the search for $\tilde{y}_i$ includes $\bar{y}_i$, since the latter is not $y_i^\star$. Thus we obtain

$$\begin{aligned}|Y_i| > 1 \;\wedge\; y_i^\star \in Y_i &\Rightarrow \widehat{c_-}(x_i, \mathcal{G}_i(\tilde{y}_i)) \le \widehat{c_+}(x_i, \mathcal{G}_i(y_i^\star))\\ &\Rightarrow f^\star(x_i; \tilde{y}_i) - f^\star(x_i; y_i^\star) - \frac{\eta_i}{2} \le (\gamma(x_i, \tilde{y}_i) + \gamma(x_i, y_i^\star)),\end{aligned}$$

as desired. □

D.2 Low Noise (Massart) Case (Theorem 6)

Fix some round i. Let $\mathcal{F}_i$ be the set of vector regressors used at round i of COAL and let $\mathcal{G}_i(y)$ be the corresponding regressors for label y. Let $\bar{y}_i \triangleq \operatorname{argmin}_y \widehat{c_+}(x_i, \mathcal{G}_i(y))$, $y_i^\star = \operatorname{argmin}_y f^\star(x_i; y)$, and $\tilde{y}_i \triangleq \operatorname{argmin}_{y \neq y_i^\star} \widehat{c_-}(x_i, \mathcal{G}_i(y))$. Assume Lemmas 8 and 9 hold. The label complexity L_2 for round i is

$$\sum_y Q_i(y) = \sum_y \mathbb{1}\left(|Y_i| > 1 \wedge y \in Y_i\right) \mathbb{1}\left(\widehat{\gamma}(x_i, y, \mathcal{F}_i) > \eta_i\right) = \sum_y \mathbb{1}\left(|Y_i| > 1 \wedge y \in Y_i\right) Q_i(y).$$

We need to do two things with $Q_i(y)$, so we have duplicated it here. First, observe that $y \in Y_i$ implies that there exists a vector regressor $f \in \mathcal{F}_i$ such that $h_f(x_i) = y$. This follows since the domination condition means that there exists $g \in \mathcal{G}_i(y)$ such that $g(x_i) \leq \min_{y' \neq y} \max_{g' \in \mathcal{G}_i(y')} g'(x_i)$. Since we are using a factored representation, we can take f to use g on the yth coordinate and use the maximizers for all the other coordinates. Moreover, $|Y_i| > 1$ implies there exists a regressor that *does not* predict y. Of course, through Lemmas 8 and 9, we know that $\mathcal{F}_i \subset \mathcal{F}_{\text{csr}}(r_{I_\beta(i)})$, and so we get the bound:

$$\mathbb{1}\left(|Y_i| > 1 \wedge y \in Y_i\right) \leq \mathbb{1}\left(\exists f, f' \in \mathcal{F}_{\text{csr}}(r_{I_\beta(i)}) \mid h_f(x_i) = y \wedge h_{f'}(x_i) \neq y\right).$$

For $y \neq y_i^\star$, we take f' to be $f^\star$ which is always in the cost-sensitive regret ball. For $y_i^\star$, we take f' to be any regressor such that $h_{f'}(x_i) = \tilde{y}_i$, which must exist in the ball if $|Y_i| > 1$. We will use these as an upper bound on $Q_i(y)$ momentarily.

Secondly, we apply Lemma 11 along with the Massart noise assumption. For $y \neq y_i^\star$

$$\begin{aligned}\mathbb{1}\left(|Y_i| > 1 \wedge y \in Y_i\right) &\leq \mathbb{1}\left(f^\star(x_i; y) - f^\star(x_i; y_i^\star) - \frac{\eta_i}{2} \leq \gamma(x_i, y) + \gamma(x_i, y_i^\star)\right)\\ &\leq \mathbb{1}\left(\tau - \frac{\eta_i}{2} \leq \gamma(x_i, y) + \gamma(x_i, y_i^\star)\right).\end{aligned}$$

Recall that we use the convention that all quantities without an explicit regressor ball use $\mathcal{F}_{\text{csr}}(r_{I_\beta(i)})$. For $y_i^\star$ we obtain the same inequality but using $\tilde{y}_i$ via Lemma 11. Together this gives the bound:

$$L_2 \leq \sum_{y \neq y_i^\star} \mathbb{1}\left(\tau - \eta_i/2 \leq \gamma(x_i, y) + \gamma(x_i, y_i^\star)\right) \times Q_i(y) + \mathbb{1}\left(\tau - \eta_i/2 \leq \gamma(x_i, \tilde{y}_i) + \gamma(x_i, y_i^\star)\right) \times Q_i(y_i^\star)$$

Let us focus on just one of these terms (say where $y \neq y_i^\star$) and consider any round i where $\tau \geq 2\eta_i$.

$$\begin{aligned}\mathbb{1}\left(\tau - \eta_i/2 \leq \gamma(x_i, y) + \gamma(x_i, y_i^\star)\right) Q_i(y) &\leq \mathbb{1}\left(\tau/2 \leq \gamma(x_i, y) + \gamma(x_i, y_i^\star)\right) Q_i(y)\\ &\leq \mathbb{1}\left(\tau/4 \leq \gamma(x_i, y)\right) Q_i(y) + \mathbb{1}\left(\tau/4 \leq \gamma(x_i, y_i^\star)\right) Q_i(y)\end{aligned}$$

Using the upper bound on $Q_i(y)$, the first term here is clearly bounded by

$$\mathbb{1}\left(\tau/4 \leq \gamma(x_i, y)\right) \mathbb{1}\left(\exists f, f' \in \mathcal{F}_{\text{csr}}(r_{I_\beta(i)}) \mid h_f(x_i) = y \wedge h_{f'}(x_i) \neq y\right) \triangleq D_i(y).$$

Fortunately, the second term is bounded in the same way, since we know that $h_{f^\star} \in \mathcal{F}_{\text{csr}}(r_{I_\beta(i)})$, the fact that some f with $h_f(x_i) = y \neq y_i^\star$ exists implies that the second term is at most $D_i(y_i^\star)$.

The last term, which involves $Q_i(y_i^\star)$ is bounded in essentially the same way, since we know that when $|Y_i| > 1$ (which is all we are considering), we know there exists two functions $f, f' \in \mathcal{F}_i$ such that $h_f(x_i) = \tilde{y}_i$ and $h_{f'}(x_i) = y_i^\star$. Thus we can bound the label complexity at round i by

$$D_i(\tilde{y}_i) + D_i(y_i^\star) + \sum_{y \neq y_i^\star} \left(D_i(y) + D_i(y_i^\star)\right) \leq K D_i(y_i^\star) + 2\sum_y D_i(y).$$

For the rounds i where $\tau < 2\eta_i$ we simply upper bound the label complexity by K.

The last step in the proof is to apply Freedman's inequality to the sequence of indicators. The conditional mean of each term is at most (for rounds i where $\tau > 2\eta_i$),

$$\mathbb{E}_i\left[KD_i(y_i^\star) + 2\sum_y D_i(y)\right] \le \frac{4r_{I_\beta(i)}}{\tau}\left[K\theta_1 + 2\theta_2\right].$$

The part involving θ_2 is straightforward and the pre-multiplier follows since we are measuring the probability of querying with a cost range parameter of $\tau/4$ and over a cost-sensitive regret ball of radius $r_{I_\beta(i)}$ in $D_i(y)$. To obtain θ_1 we use the fact that if $D_i(y_i^\star) = 1$, then certainly there exists some confused label, namely $y_i^\star$, and hence the indicator in θ_1 is also 1.

The range is $3K$ since $D_i(y) \in \{0, 1\}$ and since the terms are non-negative, the variance is at most the range times the mean. In such cases, Freedman's inequality gives

$$X \le \mathbb{E}X + 2\sqrt{R\mathbb{E}X\log(1/\delta)} + 2R\log(1/\delta) \le 2\mathbb{E}X + 3R\log(1/\delta),$$

with probability at least $1-\delta$ where X is the non-negative random variable with range R and expectation $\mathbb{E}X$. The last step is by the fact that $2\sqrt{ab} \le a + b$.

In our case, we get that with probability at least $1-\delta$,

$$\sum_{i=i^\star}^{n} KD_i(y_i^\star) + 2\sum_y D_i(y) \le \sum_{i=i^\star}^{n} \frac{8r_{I_\beta(i)}}{\tau}\left[K\theta_1 + 2\theta_2\right] + 9K\log(1/\delta).$$

Here we only consider rounds $i \ge i^\star$ where $i^\star$ is the smallest index such that $\tau < 2\eta_{i^\star}$ and $i^\star \ge i_\beta$ (Recall Fact 2). For the first $i^\star$ rounds, we will upper bound the per-round label complexity by K, so that the overall label complexity is at most

$$\begin{aligned}
&Ki^\star + \sum_{i=i^\star}^{n} \frac{8r_{I_\beta(i)}}{\tau}\left[K\theta_1 + 2\theta_2\right] + 9K\log(1/\delta)\\
&\le K\sum_{i=1}^{n} \mathbb{1}\left(\tau \le 2\eta_i\right) + Ki_\beta + + \sum_{i=i_\beta}^{n} \frac{8r_{I_\beta(i)}}{\tau}\left[K\theta_1 + 2\theta_2\right] + 9K\log(1/\delta)
\end{aligned}$$

Using our choice of $\eta_i = 1/\sqrt{i}$, the first term is at most $K\lceil 4/\tau^2\rceil$. The second term is bounded by Fact 2. The last step is to use the definition of $r_{I_\beta(i)}$ to simplify the sum. Since we are in the Massart noise case, we will set $\zeta = \tau$ in the definition of r_i in Lemma 10. Since $P_\tau = 0$ by the definition of the noise condition, this yields $r_i = 14K\Delta_i/\tau$. Substituting this choice, along with our definition of Δ_i yields

$$\begin{aligned}
\sum_{i=i_\beta}^{n} r_{I_\beta(i)} &= \frac{14\kappa n^\beta K\nu_n}{\tau}\sum_{i=i_\beta}^{n}(I_\beta(i) - 1)^{-1-\beta}\\
&\le \frac{14\kappa n^\beta K\nu_n}{\tau} \times \left(2^{(1+\beta)} \times \left(\frac{c_1}{c_2}\right)^{\frac{1+\beta}{\beta}} \sum_{i=i_\beta}^{n}(i-1)^{-1-\beta}\right)\\
&\le \frac{56(c_1/c_2)\kappa n^\beta K\nu_n}{\tau}\left[\left(\frac{c_1}{c_2}\right)^{\frac{1}{\beta}} \sum_{i=2}^{n}(i-1)^{-1}\right]\\
&\le \frac{56(c_1/c_2)\kappa n^\beta K\nu_n}{\tau}\left(\frac{c_1}{c_2}\right)^{\frac{1}{\beta}}\left(2 \times \log(n)\right).
\end{aligned}$$

Including the extra $O(K)$ term, the overall bound is

$$K\left(\lceil\frac{4}{\tau^2}\rceil + 2(c_1/c_2)^{1/\beta} + 1\right) + \frac{8 \times 56 \times 25 \times 2\kappa n^\beta K \nu_n}{\tau^2}\left(\frac{c_1}{c_2}\right)^{\frac{1}{\beta}} \log(n)[K\theta_1 + 2\theta_2] + 9K\log(1/\delta)$$
$$\leq a_0 25^{1/\beta}\left(\frac{n^\beta K \log(n)\nu_n}{\tau^2}[K\theta_1 + 2\theta_2] + \frac{K\log(1/\delta)}{\tau^2}\right),$$

where a_0 is a universal constant.

For L_1 we can use a very similar argument. First,

$$L_1 = \sum_i \mathbb{1}\left(|Y_i| > 1 \wedge \exists y \in Y_i, \widehat{\gamma}(x_i, y, \mathcal{F}_i) > \eta_i\right) \leq \sum_i \mathbb{1}\left(|Y_i| > 1 \wedge \exists y \in Y_i, \gamma(x_i, y, \mathcal{F}_{\text{csr}}(r_{I_\beta(i)})) > \eta_i/2\right).$$

This inequality is an application of Lemma 10. Now as above, we know that,

$$|Y_i| > 1 \wedge y \in Y_i \Rightarrow \exists f, f' \in \mathcal{F}_{\text{csr}}(r_{I_\beta(i)}), h_f(x_i) = y \wedge h_{f'}(x_i) \neq y,$$

since if $y \in Y_i$ then some classifier must select it, and since $|Y_i| > 1$, something else must also be selected. We also know that we can always take f' to be $f^\star$ when $y \neq y_i^\star$. For $y_i^\star$ we can always take the classifier to be the one that predicts $\tilde{y}_i$.

Moreover we also have that when $\tau \geq 2\eta_i$,

$$\begin{aligned}|Y_i| > 1 \wedge y \in Y_i &\Rightarrow f^\star(x_i; y) - f^\star(x_i; y_i^\star) - \eta_i/2 \leq \gamma(x_i, y) + \gamma(x_i, y_i^\star)\\ &\Rightarrow \tau/4 \leq \gamma(x_i, y) \vee \tau/4 \leq \gamma(x_i, y_i^\star)\end{aligned}$$

Thus, putting things together, and considering only rounds where $\tau \geq 2\eta_i$ we get

$$L_1 \leq \sum_{i=1}^n \mathbb{1}\left(\tau < 2\eta_i\right) + \sum_{i=1}^n \mathbb{1}\left(\exists y \mid \exists f, f' \in \mathcal{F}_{\text{csr}}(r_{I_\beta(i)}), h_f(x_i) = y \wedge h_{f'}(x_i) \neq y \wedge \gamma(x, y) \geq \tau/4\right).$$

Here we seemingly dropped the $\gamma(x; y_i^\star) \geq \tau/4$ term from consideration since $y_i^\star$ is always in Y_i and hence the term gets included in the existential quantifier when the chosen label $y = y_i^\star$. Now we may apply Freedman's inequality to upper bound L_1 by

$$L_1 \leq i_\beta + \lceil 4/\tau^2\rceil + 2\sum_{i=i_\beta}^n \frac{4r_{I_\beta(i)}}{\tau}\theta_1 + 2\log(1/\delta) \leq a_0 25^{1/\beta}\left(\frac{n^\beta K \log(n)\nu_n}{\tau^2}\theta_1 + \frac{\log(1/\delta)}{\tau^2}\right),$$

where a_0 is a universal constant.

D.3 High noise case (Theorem 5)

Fix some round i. Let $\mathcal{F}_i$ be the set of vector regressors used at round i of COAL and let $\mathcal{G}_i(y)$ be the corresponding regressors for label y. Let $\bar{y}_i \triangleq \text{argmin}_y \widehat{c_+}(x_i, \mathcal{G}_i(y))$, $y_i^\star = \text{argmin}_y f^\star(x_i; y)$, and $\tilde{y}_i \triangleq \text{argmin}_{y \neq y_i^\star} \widehat{c_-}(x_i, \mathcal{G}_i(y))$. Assume Lemmas 8 and 9 hold. The label complexity L_2 for round i is

$$\sum_y Q_i(y) = \sum_y \mathbb{1}\left(|Y_i| > 1 \wedge y \in Y_i\right)\mathbb{1}\left(\widehat{\gamma}(x_i, y, \mathcal{F}_i) > \eta_i\right)$$

First we apply Lemma 10 on the latter indicator to get

$$\sum_y \mathbb{1}\left(|Y_i| > 1 \wedge y \in Y_i\right) \mathbb{1}\left(\gamma(x_i, y, \mathcal{F}_{\text{csr}}(r_{I_\beta(i)})) \geq \eta_i/2\right).$$

For the former indicator, observe that $y \in Y_i$ implies that there exists a vector regressor $f \in \mathcal{F}_i$ such that $h_f(x_i) = y$. This follows since the domination condition means that there exists $g \in \mathcal{G}_i(y)$ such that $g(x_i) \leq \min_{y'} \max_{g' \in \mathcal{G}_i(y')} g'(x_i)$. Since we are using a factored representation, we can take f to use g on the yth coordinate and use the maximizers for all the other coordinates.

Since $y \in Y_i$ implies there exists $f \in \mathcal{F}_i$ such that $h_f(x_i) = y$, and by Lemmas 8 and 9, we get that $f \in \mathcal{F}_{\text{csr}}(r_{I_\beta(i)})$. Similarly there exists $f' \in \mathcal{F}_i$ such that $h_{f'}(x_i) \neq y$. Thus we can bound the the label complexity for round i as,

$$\begin{aligned}
&\sum_y \mathbb{1}\left(\exists f, f' \in \mathcal{F}_{\text{csr}}(r_{I_\beta(i)}) \mid h_f(x_i) = y \neq h_{f'}(x_i)\right) \mathbb{1}\left(\gamma(x_i, y, \mathcal{F}_{\text{csr}}(r_{I_\beta(i)})) \geq \eta_i/2\right) \\
&= \sum_y \mathbb{1}\left(x \in \text{DIS}(r_{I_\beta(i)}, y) \wedge \gamma(x_i, y, \mathcal{F}_{\text{csr}}(r_{I_\beta(i)})) \geq \eta_i/2\right).
\end{aligned}$$

Now we can apply Freedman's inequality on the sequence here to find that with probability at least $1 - \delta$,

$$L_2 \leq K i_\beta + \sum_{i=i_\beta}^{n} \frac{4 r_{I_\beta(i)}}{\eta_i} \theta_2 + 3K \log(1/\delta)$$

Again $i_\beta = 2(c_1/c_2)^{1/\beta} + 1$ is from Fact 2. We just need to bound upper bound the sequence

$$\begin{aligned}
\sum_{i=i_\beta}^{n} \frac{r_{I_\beta(i)}}{\eta_i} &= 2 \sum_{i=i_\beta}^{n} \sqrt{i} \sqrt{\frac{14K\kappa n^\beta \nu_n}{(I_\beta(i) - 1)^{1+\beta}}} \\
&\leq 2\sqrt{14K\kappa n^\beta \nu_n} \times \sum_{i=i_\beta}^{n} \sqrt{\frac{2^{1+\beta} i}{(c_2/c_1)^{\frac{1+\beta}{\beta}} (i-1)^{1+\beta}}} \\
&\leq 2\sqrt{14K\kappa n^\beta \nu_n} \times \sum_{i=i_\beta}^{n} \sqrt{\frac{2^{2+\beta}}{(c_2/c_1)^{\frac{1+\beta}{\beta}} (i-1)^{\beta}}} \\
&\leq \sqrt{448(c_1/c_2)^{\frac{1+\beta}{\beta}} K\kappa n^\beta \nu_n} \times \sum_{i=1}^{n-1} i^{-\beta/2} \\
&\leq 2\sqrt{448(c_1/c_2)^{\frac{1+\beta}{\beta}} K\kappa n^\beta \nu_n} \times n^{1-\beta/2} \\
&\leq 2\sqrt{448(c_1/c_2)^{\frac{1+\beta}{\beta}} K\kappa \nu_n} \times n.
\end{aligned}$$

The first line follows by the definition of η_i and by optimizing the bound in Lemma 9 using the definition of Δ_i. The second line uses Fact 2. The remaining steps are simple calculations using $\beta \in (0, 1)$ and an integral bound.

Thus in total we get a label complexity of

$$L_2 \leq a_0 (25)^{1/\beta} \left(n\theta_2 \sqrt{K\nu_n} + K \log(1/\delta)\right).$$

Similarly for L_1 we can derive the bound

$$L_1 \leq \sum_i \mathbb{1}\left(\exists y \mid \gamma(x_i, y, \mathcal{F}_{\text{csr}}(r_{I_\beta(i)})) \geq \eta_i/2 \wedge x \in \text{DIS}(r_{I_\beta(i)}, y)\right).$$

and then apply Freedman's inequality to this sequence to obtain that with probability at least $1-\delta$

$$L_1 \leq i_\beta + 2\sum_{i=i_\beta}^{n} \frac{2r_{I_\beta}}{\eta_i}\theta_1 + 3\log(1/\delta) \leq a_0(25)^{1/\beta}\left(n\theta_1\sqrt{K\nu_n} + \log(1/\delta)\right).$$

References

[1] A. Agarwal. Selective sampling algorithms for cost-sensitive multiclass prediction. In *International Conference on Machine Learning*, 2013.

[2] 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. In *International Conference on Machine Learning*, 2014.

[3] M.-F. Balcan and P. Long. Active and passive learning of linear separators under log-concave distributions. In *Conference on Learning Theory*, 2013.

[4] M.-F. Balcan, A. Beygelzimer, and J. Langford. Agnostic active learning. In *International Conference on Machine Learning*, 2006.

[5] M.-F. Balcan, A. Broder, and T. Zhang. Margin based active learning. In *Conference on Learning Theory*, 2007.

[6] A. Beygelzimer, S. Dasgupta, and J. Langford. Importance weighted active learning. In *International Conference on Machine Learning*, 2009.

[7] A. Beygelzimer, D. Hsu, J. Langford, and T. Zhang. Agnostic active learning without constraints. In *Advances in Neural Information Processing Systems*, 2010.

[8] A. Beygelzimer, J. Langford, L. Li, L. Reyzin, and R.E. Schapire. Contextual bandit algorithms with supervised learning guarantees. In *Artificial Intelligence and Statistics*, 2011.

[9] R.M. Castro and R.D. Nowak. Minimax bounds for active learning. *Transaction on Information Theory*, 2008.

[10] G. Cavallanti, N. Cesa-Bianchi, and C. Gentile. Learning noisy linear classifiers via adaptive and selective sampling. *Machine Learning*, 2011.

[11] K.-W. Chang, A. Krishnamurthy, A. Agarwal, H. Daumé III, and J. Langford. Learning to search better than your teacher. In *International Conference on Machine Learning*, 2015.

[12] S. Dasgupta, D. Hsu, and C. Monteleoni. A general agnostic active learning algorithm. In *Advances in Neural Information Processing Systems*, 2007.

[13] H. Daumé III, J. Langford, and D. Marcu. Search-based structured prediction. *Machine Learning*, 2009.

[14] O. Dekel, C. Gentile, and K. Sridharan. Robust selective sampling from single and multiple teachers. In *Conference on Learning Theory*, 2010.

[15] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. In *Conference on Learning Theory*, 2010.

[16] S. Hanneke. Theory of disagreement-based active learning. *Foundations and Trends in Machine Learning*, 2014.

[17] 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*, 2015.

[18] N. Karampatziakis and J. Langford. Online importance weight aware updates. In *Uncertainty in Artificial Intelligence*, 2011.

[19] D.D. Lewis, Y. Yang, T.G. Rose, and F. Li. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 2004. Data available at http://www.jmlr.org/papers/volume5/lewis04a/lyrl2004_rcv1v2_README.htm.

[20] N. Littlestone and M.K. Warmuth. The weighted majority algorithm. In *Foundations of Computer Science*, 1989.

[21] P. Massart and É. Nédélec. Risk bounds for statistical learning. *The Annals of Statistics*, 2006.

[22] F. Orabona and N. Cesa-Bianchi. Better algorithms for selective sampling. In *International Conference on Machine Learning*, 2011.

[23] S. Ross and J.A. Bagnell. Reinforcement and imitation learning via interactive no-regret learning. *arXiv:1406.5979*, 2014.

[24] S. Ross, P. Mineiro, and J. Langford. Normalized online learning. In *Uncertainty in Artificial Intelligence*, 2013.

[25] B. Settles. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 2012.

[26] T. Shi, J. Steinhardt, and P. Liang. Learning where to sample in structured prediction. In *Artificial Intelligence and Statistics*, 2015.

[27] C.N. Silla Jr and A.A. Freitas. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 2011.

[28] V. Syrgkanis, A. Krishnamurthy, and R.E. Schapire. Efficient algorithms for adversarial contextual learning. In *International Conference on Machine Learning*, 2016.

[29] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S.E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition*, 2015.

[30] C. Zhang and K. Chaudhuri. Beyond disagreement-based agnostic active learning. In *Advances in Neural Information Processing Systems*, 2014.