Appendix to: The Offset Tree for Learning with Partial Labels

1 Definitions

A *cost-sensitive* k-class classification problem is defined by a distribution D over $X \times [0,1]^k$, where X is the observation space. The goal is to find a classifier $h : X \to \{1, ..., k\}$ minimizing the expected loss $l(h, D) = E_{(x,\vec{c})\sim D}\left[c_{h(x)}\right]$, where $\vec{c} \in [0,1]^k$ is cost of the k choices. The cost sensitive regret of any classifier h is defined as the difference between the loss rate and the minimum possible loss rate, $\text{reg}_\eta(h, D) = l(h, D) - \min_{h'} l(h', D)$.

In the supervised learning setting, the cost-sensitive learner receives training examples with a complete specification of costs for all classes (or actions). In the partial feedback setting, only the cost of one class is revealed. Which cost is revealed is either chosen by the learner (after observing the side information), or by the world (drawn from some known distribution over actions given the example). For simplicity, all the algorithms presented here use the uniform distribution over actions but can be easily modified to use any other distribution (with the corresponding modifications to the bounds).

Multiclass classification is a special case of cost-sensitive classification where for every $x \in X$ all costs are 1, except for one class $y \in \{1, \ldots, k\}$ whose cost is 0. The distribution can be redefined to be over $X \times \{1, \ldots, k\}$.

Importance-weighted k-class classification is essentially k-class classification where each example has an associated weight specifying how important it is to predict its label correctly. Formally, the problem is defined by a distribution D on $X \times \{1, \ldots, k\} \times [0, \infty]$ and loss function $E_{(x,y,w)\sim D}\left[wI(b(x) \neq y)\right]$,

Binary classification is a restriction to 2-class classification where the loss is 0 for one class and 1 for the other. In other words, the binary error rate is $e(h, D) = \mathbf{Pr}_{(x,y)\sim D}(h(x) \neq y)$ and binary regret is $\text{reg}_e(h, D) = e(h, D) - \min_{h'} e(h', D)$.

Squared error regression uses datasources with a label y in the interval $[0, 1]$. The squared loss rate of any regressor $h : X \to [0, 1]$ is defined as $s(h, D) = E_{(x,y)\sim D}(y - h(x))^2$ and the squared error regret is defined as $\text{reg}_s(h, D) = s(h, D) - \min_{h'} s(h', D)$.

2 Proof of Main Results

We follow the notations in the paper.

Theorem 2.1. (Binary Offset Regret) For all 2-class partial label problems D and all binary classifiers c,

$$\text{reg}_\eta(c, D) \leq \text{reg}_e(c, \text{BO}(D)).$$

For the proof, we use the notion of *importance weighted regret*. For a classifier $c : X \to Y$ and an importance weighted distribution P over $X \times Y \times [0, \infty)$, the importance weighted regret of c on P is defined as $\text{regret}(c, P) = \ell(c, P) - \min_{c'} \ell(c', P)$.

Proof: The proof has two steps. We first bound the partial label regret of c in terms of importance weighted regret, and then apply known results to relate the importance weighted regret to binary regret.

Conditioned on a particular value of x, we either make a mistake or we do not. If no mistake is made, then the regrets of both sides are 0. If a mistake is made, assume without loss of generality that choice 1 has the smaller expected cost and choice -1 is chosen. For a set A, let $U(A)$ denote an element chosen uniformly at random from A. (This notation is also used in later sections.) The expected importance weight of choice -1 is given by

$$\begin{aligned}&\mathbf{E}_{\vec{r}\sim D|x}\mathbf{E}_{a\in U(\{1,-1\})}\left[2\left|r_a-1/2\right|\cdot\mathbf{1}\left(a(r_a-1/2)<0\right)\right]\\&=\mathbf{E}_{\vec{r}\sim D|x}\left[\left(\frac{1}{2}-r_1\right)_+ + \left(r_{-1}-\frac{1}{2}\right)_+\right]\end{aligned}$$

where we use the operators $(Z)_+ = Z\cdot\mathbf{1}(Z>0)$, and $\mathbf{1}(\cdot)$ is the indicator function which is 1 when its argument is true, and to 0 otherwise. The difference in expected importance weights between label 1 and label -1 is:

$$\begin{aligned}&\mathbf{E}_{\vec{r}\sim D|x}\left[\left(\frac{1}{2}-r_{-1}\right)_+ + \left(r_1-\frac{1}{2}\right)_+\right]\\&-\mathbf{E}_{\vec{r}\sim D|x}\left[\left(\frac{1}{2}-r_1\right)_+ + \left(r_{-1}-\frac{1}{2}\right)_+\right]\\=&\mathbf{E}_{\vec{r}\sim D|x}\left[\left(\frac{1}{2}-r_{-1}\right) + \left(r_1-\frac{1}{2}\right)\right]\\=&\mathbf{E}_{\vec{r}\sim D|x}[r_1-r_{-1}]=\mathrm{reg}_\eta(c,D|x).\end{aligned}$$

This shows that the importance weighted regret of the binary classifier is the policy regret. To relate the importance weighted regret to the binary regret, a folk theorem (see [23]) says that importance-weighted regret is bounded by binary regret times the expected importance. The expected importance is $\mathbf{E}_{\vec{r}\sim D|x}\mathbf{E}_{a\in U(\{1,-1\})}2|r_a-1/2|$. Since the importance is always bounded by 1 this gives us the theorem. ∎

Theorem 2.2. (Offset Tree Regret) *For all k-class partial label problems D, for all binary classifiers c,*

$$\begin{aligned}&\mathrm{reg}_\eta(\mathrm{OT}_c,D)\\\le&\,\mathrm{reg}_e(c,\mathrm{OT}(D))\mathbf{E}_{(x,\vec{r})\sim D}\sum_{n(a,a')\in T}\left[\,|r_a-1/2|+|r_{a'}-1/2|\,\right]\\\le&(k-1)\,\mathrm{reg}_e(c,\mathrm{OT}(D)),\end{aligned}$$

where $n(a,a')$ ranges over the $(k-1)$ internal nodes in T, and a and a' are its inputs determined by c's predictions.

Proof: We fix x, taking the expectation over the draw of x at the end. The first step of the proof is to show that the partial label regret is bounded by the sum of the importance-weighted regrets over the binary prediction problems in the tree. We then apply the costing analysis [23] to bound the latter in terms of binary classification regret.

We prove the first step by induction on the nodes in the tree. The induction hypothesis is that the sum of the importance weighted regrets of the nodes in any subtree bounds the regret of the output choice for the subtree. The hypothesis trivially holds for one-node trees.

Consider a node n making an importance weighted decision between choices a and a' provided by subtrees L and R. The expected importance of choice a is given by:

$$\begin{aligned}&\mathbf{E}_{\vec{r}\sim D}\frac{1}{2}2(r_a-1/2)_+ + \frac{1}{2}2(1/2-r_{a'})_+\\=&\mathbf{E}_{\vec{r}\sim D}(r_a-1/2)_+ + (1/2-r_{a'})_+\end{aligned}$$

since a and a' each have probability $1/2$ of being drawn, conditioned on the action reaching the node. (It is important to note here that for internal nodes *only* two actions can generate examples for those nodes because the classifier at leaf-wards nodes can only agree with one action.)

Without loss of generality, assume that a has a larger expected reward.

The expected importance-weighted binary regret of this decision, denoted by regret_n is either 0 if the classifier predicts a or

$$\begin{aligned}\mathrm{regret}_n =& \mathbf{E}_{\vec{r}\sim D|x}\left[(r_a - 1/2)_+ + (1/2 - r_{a'})_+\right] \\ & - \mathbf{E}_{\vec{r}\sim D|x}\left[(r_{a'} - 1/2)_+ + (1/2 - r_a)_+ +\right] \\ =& \mathbf{E}_{\vec{r}\sim D|x}[1/2 - r_{a'} + r_a - 1/2] \\ =& \mathbf{E}_{\vec{r}\sim D|x}[r_a - r_{a'}]\end{aligned}$$

if the classifier predicts a'.

Let T_n be the subtree rooted at n and $\mathrm{leaves}(T_n)$ be the leaves of T_n. The regret of the choice a output by T_n is given by

$$\mathrm{Reg}_{T_n} = \max_{y\in \mathrm{leaves}(T_n)} \mathbf{E}_{\vec{r}\sim D|x}[r_y] - \mathbf{E}_{\vec{r}\sim D|x}[r_a].$$

If the best choice in $\mathrm{leaves}(T_n)$ comes from the subtree L which produces the choice a preferred by the classifier at n, we have

$$\begin{aligned}\mathrm{Reg}_{T_n} &= \max_{y\in \mathrm{leaves}(L)} \mathbf{E}_{\vec{r}\sim D|x}[r_y] - \mathbf{E}_{\vec{r}\sim D|x}[r_a] \\ &= \mathrm{Reg}_L \le \sum_{u\in L} \mathrm{regret}_u \le \sum_{u\in T_n} \mathrm{regret}_u,\end{aligned}$$

proving the induction.

On the other hand, if the best choice comes from the subtree R which does not produce a preferred by the classifier at n, we have

$$\begin{aligned}\mathrm{Reg}_{T_n} &= \max_{y\in \mathrm{leaves}(R)} \mathbf{E}_{\vec{r}\sim D|x}[r_y] - \mathbf{E}_{\vec{r}\sim D|x}[r_a] \\ &= \mathrm{Reg}_R + \mathbf{E}_{\vec{r}\sim D|x}[r_{a'}] - \mathbf{E}_{\vec{r}\sim D|x}[r_a] \\ &\le \sum_{u\in R} \mathrm{regret}_u + \mathrm{regret}_n \le \sum_{u\in T_n} \mathrm{regret}_u,\end{aligned}$$

completing the induction proof.

Let a be the choice output by the root of the tree T. The induction hypothesis implies that the regret of predicting a on x is bounded by

$$\mathrm{reg}_\eta(a, D \mid x) \le \sum_{u\in T} \mathrm{regret}_u \le (k-1)\,\mathrm{regret}(c, D \mid x),$$

where $\mathrm{regret}(c, D\,|\,x)$ is the average importance weighted binary regret of c on D over the $k-1$ nodes in T, conditioned on x.

The second step of the proof is to bound the average importance weighted binary regret in terms of binary regret. A folk theorem proved in [23] says that for all binary classifiers c and all importance-weighted binary problems P,

$$\mathrm{regret}(c, P) = \mathrm{reg}_e(c, P')\mathbf{E}_{(x,y,w)\sim P}[w],$$

where $\mathrm{reg}_e(c, P')$ is the binary regret of c on the induced binary distribution P'.

For the offset tree reduction, the expected importance of any node n deciding between actions a and a' is

$$\begin{aligned}&\mathbf{E}_{\vec{r}\sim D|x}\mathbf{E}_{a''\sim U(\{1,\ldots,k\})} k|r_{a''} - 1/2| \cdot \mathbf{1}\,(a'' = a \text{ or } a'' = a') \\ &= \mathbf{E}_{\vec{r}\sim D|x}|r_a - 1/2| + |r_{a'} - 1/2| \le 1\end{aligned}$$

Since $\mathrm{regret}(c, D|x)$ is an expectation over importance weighted regrets where the importance weights are all ≤ 1, the folk theorem gives us $\mathrm{regret}(c, D|x) \le \mathrm{reg}_e(c, D|x)$, completing the proof for any x. Take an expectation over all x finishes the proof. ∎

Theorem 2.3. *For all reductions R, R^{-1}, there exists a partial label problem D and an oracle O such that*

$$\operatorname{reg}_\eta(R^{-1}(O), D) \geq (k-1)\operatorname{reg}_e(O, R(D))$$

Proof: The proof is by construction. We choose a distribution D which places mass on k partial label example $(x, \vec{r})$ where $\vec{r}$ has one entry (chosen uniformly at random) with value 1 and the others with value 0. The features x consist of the binary representation of the index with reward 1.

The oracle O operates by simulating R. R chooses an action a according to some distribution $p(a)$ and observes the reward r_a, then makes zero or more advice calls to the oracle. In particular, R produces some simulatable distribution over advice calls when $r_a = 0$. The oracle ignores all advice calls from R and chooses to answer queries with zero error rate according to a draw from this simulatable distribution.

There are two cases: either R observes 0 reward (with probability $(k-1)/k$) or it observes reward 1 (with probability $1/k$. In the first case, the oracle has 0 error rate (and, hence 0 regret) with respect to the sequence of advice queries from R. In the second case it has error rate (and regret) 1. The average error rate (over the random draw from D) is $1/k$.

The inverse reduction R^{-1} has access to only the unlabeled example x and the oracle O. Since the oracle's answers are independent of the draw from D, the output action a has reward $r_a = 0$ with probability $(k-1)/k$ and reward 1 with probability $1/k$, implying a regret of $(k-1)/k$ with respect to the best policy. This is a factor of $k-1$ greater than the regret of the oracle, implying the proof. ▮

3 Analysis of Simple Reductions

In this section, we analyze two baseline approaches for reducing partial label problems to basic supervised learning problems. These schemes have been discussed previously, but the analysis here appears new.

3.1 The Regression Approach

A *squared error regression problem* is defined by a distribution D over $X \times \mathbb{R}$. The goal is to find a *regressor* $h : X \to \mathbb{R}$ minimizing the expected squared loss $\mathbf{E}_{(x,y)\sim D}(f(x)-y)^2$.

The simplest way to reduce a partial-feedback problem to a well known learning problem is to regress on the value of a choice as in Algorithm 1, and then use the argmax classifier as in Algorithm 2. Note that instead of learning a single regressor, a simple variant might learn a different regressor for each choice.

Algorithm 1: Partial-Regression (regression algorithm Regress, partial label dataset S)

```
Let $S' = \emptyset$
for each $(x, a, r_a) \in S$ do
    Add $((x, a), r_a)$ to $S'$.
return $s$ = Regress($S'$).
```

Algorithm 2: Argmax (regressor s, unlabeled example x)

```
return $\arg\max_{a'} s(x, a')$
```

We state the following theorem relating the regret of the resulting predictor to that of the regressor.

Theorem 3.1. *For all partial label problems D, for all squared-error regressors h,*

$$\operatorname{reg}_\eta(\text{Argmax}(s), D) \leq \sqrt{2k \operatorname{reg}_s(h, \text{Partial-Regression}(D))}.$$

The theorem has a square root, which is undesirable, because the theorem is vacuous when the right hand side is greater than 1. Removing this square root is one of the motivations for the other reductions.

Proof: Let the argmax classifier based on h choose some action a with true value $v_a = \mathbf{E}_{(x,\vec{r})\sim D}[r_a]$. Some other action a^* may have a larger expected reward $v_{a^*} > v_a$. The squared error regret suffered by h on a is $\mathbf{E}_{(x,\vec{r})\sim D}[(r_a - v_a)^2 - (r_a - h(x,a))^2] = (v_a - h(x,a))^2$. Similarly for a^* we have regret $(v_{a^*} - h(x,a^*))^2$. In order for a to be chosen over a^*, we must have $h(x,a) \geq h(x,a^*)$. Convexity of the two regrets implies that the minima is reached when $h(x,a) = h(x,a^*) = \frac{v_a + v_{a^*}}{2}$, where the regret for each of the two choices is $\left(\frac{v_{a^*} - v_a}{2}\right)^2$. The regressor need not suffer any regret on the other $k-2$ arms. Thus with average regret $\text{reg}_s = \frac{(v_{a^*} - v_a)^2}{2k}$ a regret of $v_{a^*} - v_a$ can be induced. This completes the proof since $v_{a^*} - v_a \leq \sqrt{2k\,\text{reg}_s}$. ▮

3.2 The Importance Weighted Classification Approach

Algorithm 3: PI-Train (binary classifier learner Learn, partial label dataset S)

```
Let $S' = \emptyset$
for each $(x, a, r_a) \in S$ do
    Add $(x, a, kr_a)$ to $S'$.
return All-Pairs-Train (Learn, Costing($S'$))
```

Algorithm 4: PI-Test (Binary classifier c, Unlabeled sample x)

```
return All-Pairs-Test($c, x$).
```

The PI-Train algorithm creates importance weighted samples in step 2.(b), the importance weights are stripped using Costing (which can be understood as rejection sampling according to the weight values), and then multiclass samples are reduced to binary using the All-Pairs reduction.

At test time, we make a choice using the All-Pairs algorithm for the learned binary classifier as in algorithm 4. A basic theorem applies to this approach.

Theorem 3.2. *For all partial label problems D over k choices, for all binary classifiers c,*

$$\begin{aligned}
&\text{reg}_\eta(\text{PI-Test}(c), D)\\
&\quad\leq \text{reg}_e(c, \text{PI-Train}(D))(k-1)\mathbf{E}_{(x,\vec{c})\sim D}\sum_a (1 - c_a)\\
&\quad\leq \text{reg}_e(c, \text{PI-Train}(D))(k-1)k.
\end{aligned}$$

Proof: The proof first bounds the policy regret in terms of the importance weighted multiclass regret. Then, we apply known results for the other reductions to relate the policy regret to binary classification regret.

Fix a particular x. The policy regret of choosing action a over the best action a^* is $\mathbf{E}_{r\sim D|x}[r_{a^*}] - \mathbf{E}_{r\sim D|x}[r_a]$.

The importance weighted multiclass loss of action a is $\mathbf{E}_{r\sim D|x}\sum_{a'\neq a}\frac{kr_{a'}}{k} = \mathbf{E}_{r\sim D|x}\sum_{a'\neq a} r_{a'}$ since the loss is proportional to $kr_{a'}$ with probability $\frac{1}{k}$. This implies the importance weighted regret of

$$\mathbf{E}_{r\sim D|x}\sum_{a'\neq a} r_{a'} - \mathbf{E}_{r\sim D|x}\sum_{a'\neq a^*} r_{a'} = \mathbf{E}_{r\sim D|x}[r_{a^*} - r_a],$$

which is the same as the policy regret.

Importance weighted regret is bounded by unweighted regret times the expected importance (see [23]), which in turn is bounded by k. Multiclass regret on k classes is bounded by binary regret times $k-1$ using the All-Pairs reduction [9], which completes the proof. ▮

References

[1] N. Abe, A. Biermann, and P. Long. Reinforcement learning with immediate rewards and linear hypotheses, *Algorithmica*, 37(4): 263–293, 2003.

[2] P. Auer. Using confidence bounds for exploitation-exploration tradeoffs, JMLR, 3: 397–422, 2002.

[3] Peter Auer, Nicol Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem, FOCS 322–331, 1995.

[4] P. Auer, N. Cesa-Bianchi, P. Fischer. Finite time analysis of the multi-armed bandit problem, *Machine Learning*, 47: 235–256, 2002.

[5] A. Beygelzimer, J. Langford, and P. Ravikumar. Filter trees for cost sensitive multiclass classification, 2008.

[6] A. Beygelzimer, J. Langford, and P. Ravikumar. Error Correcting Tournaments, 2008.

[7] C. Elkan. The foundations of cost-sensitive learning, IJCAI 2001, 973–978.

[8] Even-dar, E., Mannor, S., and Mansour, Y. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. JMLR 2006, 7, 1079-1105.

[9] T. Hastie and R. Tibshirani. Classification by pairwise coupling, NIPS 1997.

[10] J. Heckman. Sample Selection Bias as a Specification Error, Econometrica, Vol. 47, No. 1 (Jan., 1979), 153–161.

[11] M. Kearns, Y. Mansour, and A. Y. Ng. Approximate Planning in Large POMDPs via Reusable Trajectories, NIPS 2000.

[12] S. Kulkarni. On bandit problems with side observations and learnability, Allerton 1993, 83–92.

[13] J. Langford and A. Beygelzimer. Sensitive error correcting output codes, COLT 2005.

[14] J. Langford and T. Zhang. The Epoch-Greedy Algorithm for Contextual Multiarmed Bandits, NIPS 2007.

[15] S. Pandey, D. Agarwal, D. Chakrabati, V. Josifovski. Bandits for taxonomies: a model based approach, SDM 2007.

[16] H. Robbins. Some aspects of the sequential design of experiments, *Bulletins of the American Mathematical Society*, 58: 527–535, 1952.

[17] A. Strehl, C. Mesterham, M. Littman, and H. Hirsh, Experience-efficient learning in associative bandit problems, ICML 2006, 889–896.

[18] C. Blake and C. Merz, UCI Repository of machine learning databases. University of California, Irvine.

[19] C. C. Wang, S. Kulkarni, and H. Vincent Poor, Bandit problems with side observations, *IEEE Transactions on Automatic Control*, 50(5), 2005.

[20] I. Witten and E. Frank. Data Mining: Practical machine learning tools with Java implementations, 2000: http://www.cs.waikato.ac.nz/ml/weka/.

[21] M. Woodruff. A one-armed bandit problem with concomitant variates, JASA, 74 (368): 799–806, 1979.

[22] Bianca Zadrozny, Ph.D. Thesis, University of California, San Diego, 2003.

[23] B. Zadrozny, J. Langford, and N. Abe. Cost sensitive learning by cost-proportionate example weighting, ICDM 2003.