Online Appendix for RELEAF: An Algorithm for Learning and Exploiting Relevance

Cem Tekin, *Member, IEEE*, Mihaela van der Schaar, *Fellow, IEEE*

Abstract

This online appendix is composed of two sections. In the first section we give the proof of Theorem 5 in [1]. The second section is an extensive version of the numerical results given in Section V of [1].

I. Proof of Theorem 5

A. Preliminaries

Let $A := |\mathcal{A}|$. We first define a sequence of events which will be used in the analysis of the regret of RELEAF. For $\boldsymbol{p} \in \boldsymbol{\mathcal{P}}_{\mathcal{R}(a),t}$, let $\pi(a,\boldsymbol{p}) = \mu(a, \boldsymbol{x}^*_{\mathcal{R}(a)}(\boldsymbol{p}))$, where $\boldsymbol{x}^*_{\mathcal{R}(a)}(\boldsymbol{p}) = \{x^*_i(p_i)\}_{i\in\mathcal{R}(a)}$ such that $x^*_i(p_i)$ is the type i context at the geometric center of p. Let $W(\mathcal{R}(a))$ be the set of D_{rel}-tuple of types such that $\mathcal{R}(a) \subset \boldsymbol{w}$, for every $\boldsymbol{w} \in W(\mathcal{R}(a))$. We have $|W(\mathcal{R}(a))| = \binom{D-|\mathcal{R}(a)|}{2D_{\text{rel}}-|\mathcal{R}(a)|}$. For a D_{rel}-tuple of types $\boldsymbol{w}$, let $\mathcal{D}(\boldsymbol{w}, D')$ be the set of D'-tuple of types whose elements are from the set $\mathcal{D}_{-\boldsymbol{w}}$.

For any $\boldsymbol{w} \in W(\mathcal{R}(a))$ and $\boldsymbol{j} \in \mathcal{D}(\boldsymbol{w}, D_{\text{rel}})$, let

$$\text{INACC}_t(a,\boldsymbol{w},\boldsymbol{j}) := \left\{ |\bar{r}_t^{(\boldsymbol{w},\boldsymbol{j})}(\boldsymbol{p}_{\boldsymbol{w},t}, \boldsymbol{p}_{\boldsymbol{j},t}, a) - \pi(a, \boldsymbol{p}_{\mathcal{R}(a),t})| > \frac{3}{2} L\sqrt{D_{\text{rel}}} \max_{i\in\mathcal{R}(a)} s(\boldsymbol{p}_{\mathcal{R}(a),t}) \right\},$$

be the event that the sample mean reward of action a corresponding to the $2D_{\text{rel}}$-tuple of types $(\boldsymbol{w},\boldsymbol{j})$ is *inaccurate* for action a. Let

$$\text{ACC}_t(a) := \bigcap_{\boldsymbol{w}\in W(\mathcal{R}(a))} \bigcap_{\boldsymbol{j}\in\mathcal{D}(\boldsymbol{w},D_{\text{rel}})} \text{INACC}_t(a,\boldsymbol{w},\boldsymbol{j})^C$$

be the event that sample mean reward estimates of action a corresponding to all tuples $(\boldsymbol{w},\boldsymbol{j})$ $\boldsymbol{w} \in W(\mathcal{R}(a))$ and $\boldsymbol{j} \in \mathcal{D}(\boldsymbol{w}, D_{\text{rel}})$ are accurate. Consider $t \in \tau(T)$. Let

$$\text{WNG}_t(a) := \bigcup_{\boldsymbol{w}\in W(\mathcal{R}(a))} \{\boldsymbol{w} \notin \text{Rel}_t(a)\}$$

be the event that some D_{rel}-tuple that contains $\mathcal{R}(a)$ is not in the set of relevant tuples of types for action a. Let $\text{WNG}_t := \bigcup_{a\in\mathcal{A}} \text{WNG}_t(a)$, and $\text{CORR}_T := \bigcap_{t\in\tau(T)} \text{WNG}_t^C$, be the event that all D_{rel}-tuples of

types that contain the set of relevant contexts of each action is an element of the set of candidate relevant D_{rel}-tuples types corresponding to that action at all exploitation steps.

We first prove several lemmas related to Theorem 5. The next lemma gives a lower bound on the probability of CORR_T.

Lemma 1. *For RELEAF, for all $a \in \mathcal{A}$, $t \in \tau(T)$, we have $\text{P}(\textit{INACC}_t(a, \boldsymbol{w}, \boldsymbol{j})) \leq \frac{2\delta}{AD^* t^4}$. for all $\boldsymbol{w} \in W(\mathcal{R}(a))$, $\boldsymbol{j} \in \mathcal{D}(\boldsymbol{w}, D_{rel})$, and $\text{P}(\textit{CORR}_T) \geq 1 - \delta$ for any T.*

Proof: For $t \in \tau(T)$, we have $\mathcal{U}_t = \emptyset$, hence

$$S_t^{\boldsymbol{v}(\boldsymbol{q})}(\boldsymbol{q}, a) \geq \frac{2\log(tAD^*/\delta)}{(L \min_{i \in \boldsymbol{v}(\boldsymbol{q})} s(p_{i,t}))^2},$$

for all $a \in \mathcal{A}$, $\boldsymbol{q} \in Q(t)$. Due to the *Similarity Assumption*, since for all $a \in \mathcal{A}$, $\boldsymbol{w} \in W(\mathcal{R}(a))$ and $\boldsymbol{j} \in \mathcal{D}(\boldsymbol{w}, D_{\text{rel}})$ the rewards in $\bar{r}_t^{(\boldsymbol{w},\boldsymbol{j})}((\boldsymbol{p}_{\boldsymbol{w},t}, \boldsymbol{p}_{\boldsymbol{j},t}), a)$ are sampled from distributions with mean between $[\pi(a, \boldsymbol{p}_{\mathcal{R}(a),t}) - \frac{L\sqrt{D_{\text{rel}}}}{2} \max_{i \in \mathcal{R}(a)} s(p_{i,t}), \pi(a, \boldsymbol{p}_{\mathcal{R}(a),t}) + \frac{L\sqrt{D_{\text{rel}}}}{2} \max_{i \in \mathcal{R}(a)} s(p_{i,t})]$, using a Chernoff bound we get

$$\begin{aligned}
&\text{P}(\text{INACC}_t(a, \boldsymbol{w}, \boldsymbol{j})) \\
&\leq 2\exp\left(-2(L\sqrt{D_{\text{rel}}} \max_{i \in \mathcal{R}(a)} s(p_{i,t}))^2 \frac{2\log(tAD^*/\delta)}{(L \min_{i \in (\boldsymbol{w},\boldsymbol{j})} s(p_{i,t}))^2}\right) \\
&\leq 2\delta/(AD^* t^4).
\end{aligned}$$

We have

$$\text{WNG}_t(a) \subset \bigcup_{\boldsymbol{w} \in W(\mathcal{R}(a))} \bigcup_{\boldsymbol{j} \in \mathcal{D}(\boldsymbol{w}, D_{\text{rel}})} \text{INACC}_t(a)^C.$$

Since the number of $2D_{\text{rel}}$-tuples that contain $\mathcal{R}(a)$ is $\binom{D-\mathcal{R}(a)}{2D_{\text{rel}}-\mathcal{R}(a)}$, which is less than or equal to $D^* = \binom{D-1}{2D_{\text{rel}}-1}$ since $1 \leq \mathcal{R}(a) \leq D_{\text{rel}}$, we have

$$\text{P}(\text{WNG}_t(a)) \leq 2\delta/(At^4),$$

and

$$\text{P}(\text{WNG}_t) \leq 2\delta/t^4.$$

This implies that

$$\text{P}(\text{CORR}_T^C) \leq \sum_{t \in \tau(T)} \text{P}(\text{WNG}_t)$$

$$\leq \sum_{t\in\tau(T)} \frac{2\delta}{t^4} \leq \sum_{t=3}^{\infty} \frac{2\delta}{t^4} \leq \delta.$$

■

Lemma 2. *When CORR$_T$ happens we have for all $t \in \tau(T)$*

$$|\bar{r}_t^{\hat{c}_t(a)}(\boldsymbol{p}_{\hat{c}_t(a),t}, a) - \mu(a, \boldsymbol{x}_{\mathcal{R}(a),t})| \leq 3L\sqrt{D_{rel}}(\max_{i\in\hat{c}_t(a)} s(p_{i,t}) + \max_{i\in\boldsymbol{w}} s(p_{i,t})) + 2L\sqrt{D_{rel}} \max_{i\in\mathcal{R}(a)} s(p_{i,t}).$$

Proof: From Lemma 1, CORR$_T$ happens when

$$|\bar{r}_t^{\boldsymbol{w}}(\boldsymbol{p}_{\boldsymbol{w},t}, a) - \pi(a, \boldsymbol{p}_{\mathcal{R}(a),t})| \leq \frac{3L\sqrt{D_{\text{rel}}}}{2} \max_{i\in\mathcal{R}(a)} s(p_{i,t}),$$

for all $a \in \mathcal{A}$, $\boldsymbol{w} \in W(\mathcal{R}(a))$, $t \in \tau(T)$. Since

$$|\mu(a, \boldsymbol{x}_{\mathcal{R}(a),t}) - \pi(a, \boldsymbol{p}_{\mathcal{R}(a),t})| \leq \frac{L\sqrt{D_{\text{rel}}}}{2} \max_{i\in\mathcal{R}(a)} s(p_{i,t})$$

by the Similarity Assumption, we have

$$|\bar{r}_t^{\boldsymbol{w}}(\boldsymbol{p}_{\boldsymbol{w},t}, a) - \mu(a, \boldsymbol{x}_{\mathcal{R}(a),t})| \leq 2L\sqrt{D_{\text{rel}}} \max_{i\in\mathcal{R}(a)} s(p_{i,t}), \tag{A.1}$$

for all $a \in \mathcal{A}$, $\boldsymbol{w} \in W(\mathcal{R}(a))$, $t \in \tau(T)$. Consider $\hat{c}_t(a)$. Since it is chosen from Rel$_t(a)$ as the D_{rel}-tuple of types with the minimum variation, we have on the event CORR$_T$

$$|\bar{r}_t^{(\hat{c}_t(a),\boldsymbol{k})}((\boldsymbol{p}_{\hat{c}_t(a),t}, p_{\boldsymbol{k},t}), a) - \bar{r}_t^{(\hat{c}_t(a),\boldsymbol{j})}((\boldsymbol{p}_{\hat{c}_t(a),t}, p_{\boldsymbol{j},t}), a)| \leq 3L\sqrt{D_{\text{rel}}} \max_{i\in\hat{c}_t(a)} s(p_{i,t}),$$

for all $\boldsymbol{j}, \boldsymbol{k} \in \mathcal{D}(\hat{c}_t(a), D_{\text{rel}})$. For any $\boldsymbol{w} \in W(\mathcal{R}(a))$, let $\boldsymbol{g}(\boldsymbol{w}, \hat{c}_t(a))$ be a $2D_{\text{rel}}$-tuple such that for all $i \in \boldsymbol{w}$ and $j \in \hat{c}_t(a)$, $i, j \in \boldsymbol{g}(\boldsymbol{w}, \hat{c}_t(a))$. The existence of at least one such $2D_{\text{rel}}$-tuple of types is guaranteed since $\boldsymbol{w}$ and $\hat{c}_t(a)$ are both D_{rel}-tuples of types. Hence, we have for any $\boldsymbol{w} \in W(\mathcal{R}(a))$

$$\begin{aligned}
&|\bar{r}_t^{\boldsymbol{w}}(\boldsymbol{p}_{\boldsymbol{w},t}, a) - \bar{r}_t^{\hat{c}_t(a)}(\boldsymbol{p}_{\hat{c}_t(a),t}, a)| \\
&\leq \max_{\boldsymbol{k}\in\mathcal{D}(\boldsymbol{w},D_{\text{rel}}),\boldsymbol{j}\in\mathcal{D}(\hat{c}_t(a),D_{\text{rel}})} \left\{|\bar{r}_t^{(\boldsymbol{w},\boldsymbol{k})}((\boldsymbol{p}_{\boldsymbol{w},t}, p_{\boldsymbol{k},t}), a) - \bar{r}_t^{(\hat{c}_t(a),\boldsymbol{j})}((\boldsymbol{p}_{\hat{c}_t(a),t}, p_{\boldsymbol{j},t}), a)|\right\} \\
&\leq \max_{\boldsymbol{k}\in\mathcal{D}(\boldsymbol{w},D_{\text{rel}}),\boldsymbol{j}\in\mathcal{D}(\hat{c}_t(a),D_{\text{rel}})} \Big\{|\bar{r}_t^{(\boldsymbol{w},\boldsymbol{k})}((\boldsymbol{p}_{\boldsymbol{w},t}, \boldsymbol{p}_{\boldsymbol{k},t}), a) - \bar{r}_t^{\boldsymbol{g}(\boldsymbol{w},\hat{c}_t(a))}(\boldsymbol{p}_{\boldsymbol{g}(\boldsymbol{w},\hat{c}_t(a)),t}, a)| \\
&+|\bar{r}_t^{\boldsymbol{g}(\boldsymbol{w},\hat{c}_t(a))}(\boldsymbol{p}_{\boldsymbol{g}(\boldsymbol{w},\hat{c}_t(a)),t}, a) - \bar{r}_t^{(\hat{c}_t(a),\boldsymbol{j})}((\boldsymbol{p}_{\hat{c}_t(a),t}, \boldsymbol{p}_{\boldsymbol{j},t}), a)|\Big\} \\
&\leq 3L\sqrt{D_{\text{rel}}}(\max_{i\in\hat{c}_t(a)} s(p_{i,t}) + \max_{i\in\boldsymbol{w}} s(p_{i,t}))
\end{aligned} \tag{A.2}$$

Combining (A.1) and (A.2), we get

$$|\bar{r}_t^{\hat{c}_t(a)}(\boldsymbol{p}_{\hat{c}_t(a),t}, a) - \mu(a, \boldsymbol{x}_{\mathcal{R}(a),t})| \leq 3L\sqrt{D_{\text{rel}}}(\max_{i\in\hat{c}_t(a)} s(p_{i,t}) + \max_{i\in\mathcal{D}} s(p_{i,t})) + 2L\sqrt{D_{\text{rel}}} \max_{i\in\mathcal{R}(a)} s(p_{i,t}).$$

■

B. Regret bound for exploitations

Since for $t \in \tau(T)$, $\alpha_t = \arg\max_{a \in \mathcal{A}} \bar{r}_t^{\hat{c}_t(a)}(p_{\hat{c}_t(a),t}, a)$, using the result of Lemma 2, we conclude that

$$\mu_t(\alpha_t) \geq \mu_t(a^*(\boldsymbol{x}_t)) - 6L\sqrt{D_{\text{rel}}}(\max_{i \in \hat{c}_t(a)} s(p_{i,t}) + \max_{i \in \mathcal{D}} s(p_{i,t})) - 4L\sqrt{D_{\text{rel}}} \max_{i \in \mathcal{R}(a)} s(p_{i,t}),$$

Thus, the regret in exploitation steps is bounded above by

$$\begin{aligned}
&6L\sqrt{D_{\text{rel}}} \sum_{t \in \tau(T)} (\max_{i \in \hat{c}_t(a)} s(p_{i,t}) + \max_{i \in \mathcal{D}} s(p_{i,t})) + 4L\sqrt{D_{\text{rel}}} \sum_{t \in \tau(T)} \max_{i \in \mathcal{R}(a)} s(p_{i,t}) \\
&\leq 16L\sqrt{D_{\text{rel}}} \sum_{t \in \tau(T)} \max_{i \in \mathcal{D}} s(p_{i,t}) \\
&\leq 16L\sqrt{D_{\text{rel}}} \sum_{t \in \tau(T)} \sum_{i \in \mathcal{D}} s(p_{i,t}) \\
&\leq 16LD\sqrt{D_{\text{rel}}} \max_{i \in \mathcal{D}} \left(\sum_{t \in \tau(T)} s(p_{i,t}) \right).
\end{aligned}$$

We know that as time goes on RELEAF uses partitions with smaller and smaller intervals, which reduces the regret in exploitations. In order to bound the regret in exploitations for any sequence of context arrivals, we assume a worst case scenario, where context vectors arrive such that at each t, the active interval that contains the context of each type has the maximum possible length. This happens when for each type i contexts arrive in a way that all level l intervals are split to level $l+1$ intervals, before any arrivals to these level $l+1$ intervals happen, for all $l = 0, 1, 2, \ldots$. This way it is guaranteed that the length of the interval that contains the context for each $t \in \tau(T)$ is maximized. Let $l_{\max}$ be the level of the maximum level interval in $\mathcal{P}_i(T)$. For the worst case context arrivals we must have

$$\sum_{l=0}^{l_{\max}-1} 2^l 2^{\rho l} < T \Rightarrow l_{\max} < 1 + \log_2 T/(1+\rho),$$

since otherwise maximum level hypercube will have level larger than $l_{\max}$. Hence, we have

$$\begin{aligned}
&16LD\sqrt{D_{\text{rel}}} \max_{i \in \mathcal{D}} \left(\sum_{t \in \tau(T)} s(p_{i,t}) \right) \\
&\leq 16LD\sqrt{D_{\text{rel}}} \sum_{l=0}^{1+\log_2 T/(1+\rho)} 2^l 2^{\rho l} 2^{-l} \\
&= 16LD\sqrt{D_{\text{rel}}} \sum_{l=0}^{1+\log_2 T/(1+\rho)} 2^{\rho l} \\
&\leq 16LD\sqrt{D_{\text{rel}}} 2^{2\rho} T^{\rho/(1+\rho)}.
\end{aligned}$$

Hence, we have $R_I(T) = \tilde{O}(T^{\rho/(1+\rho)})$ with probability $1-\delta$.

C. Regret bound for explorations

Recall that time t is an exploitation step only if $\mathcal{U}_t = \emptyset$. In order for this to happen we need $S_t^{\boldsymbol{v}(\boldsymbol{q})}(\boldsymbol{q}, a) \geq D_{i,t}$ for all $\boldsymbol{q} \in Q_i(t)$. The number of distinct $2D_{\text{rel}}$-tuples of types is $\binom{D}{2D_{\text{rel}}}$. Whenever action a is explored, all the counters for these $\binom{D}{2D_{\text{rel}}}$ type tuples are updated for the $2D_{\text{rel}}$-tuples of intervals that contain types of contexts present at time t, i.e. $\boldsymbol{q} \in Q_t$. Now consider a hypothetical scenario in which instead of updating the counters of all $\boldsymbol{q} \in Q_t$, the counter of only one of the randomly selected $2D_{\text{rel}}$-tuple of intervals is updated. Clearly, the exploration regret of this hypothetical scenario upper bounds the exploration regret of the original scenario. In this scenario for any $\boldsymbol{q} \in Q_t$, we have

$$S_t^{\boldsymbol{v}(\boldsymbol{q})}(\boldsymbol{q}, a) \leq \frac{2\log(tAD^*/\delta)}{(L \min_{i \in \boldsymbol{v}(\boldsymbol{q})} s(p_i))^2} + 1.$$

We fix a $2D_{\text{rel}}$-tuple of types $\boldsymbol{j} = (j_1, j_2, \ldots, j_{2D_{\text{rel}}})$, and analyze the worst-case regret due to exploration of this tuple of types, which is denoted by $R_{O,\boldsymbol{j}}(T)$. Since there are $\binom{D}{2D_{\text{rel}}}$ of such tuples of types, an upper bound on the exploration regret is $\binom{D}{2D_{\text{rel}}} R_{O,\boldsymbol{j}}(T)$.

Let $l_{\max}$ be the maximum possible level for an active interval for type i by time T. We must have $\sum_{l=0}^{l_{\max}-1} 2^{\rho l} < T$, which implies that $l_{\max} < 1 + \log_2 T/\rho$. Let $\gamma = 1 + \log_2 T/\rho$.

First, we will consider the exploration regret incurred in all configurations where type j_n's intervals has levels l_n, for $n = 1, 2, \ldots, 2D_{\text{rel}}$ such that $l_1 \leq l_2 \leq \ldots \leq l_{2D_{\text{rel}}}$. We denote this ordering by $\boldsymbol{j}^*$ and the exploration regret in this ordering by $R_{O,\boldsymbol{j}^*}(T)$. There are $(2D_{\text{rel}})!$ different configurations in which the orderings of levels of the intervals of the types are different.

Let $z = 2D_{\text{rel}}$. Consider the tuple of intervals $(p_{j_1^*}, \ldots, p_{j_{2D_{\text{rel}}}^*})$. The exploration regret for this tuple of intervals is bounded by

$$(c_O + 1)\left(2\log(TAD^*/\delta)/(2^{-2l_z}L^2) + 1\right).$$

Hence, we have

$$\begin{aligned}
&R_{O,\boldsymbol{j}^*}(T) \leq (c_O + 1) \\
&\times \sum_{l_1=0}^{\gamma} 2^{l_1} \sum_{l_2=l_1}^{\gamma} 2^{l_2} \ldots \sum_{l_z=l_{z-1}}^{\gamma} 2^{l_z} \left(\frac{2\log(TAD^*/\delta)}{2^{-2l_z}L^2} + 1\right) \\
&\leq (c_O + 1) \sum_{l_z=l_{z-1}}^{\gamma} 2^{l_z} \ldots \sum_{l_{z-1}=l_{z-2}}^{\gamma} 2^{l_{z-1}} O(T^{3/\rho}\log T) \\
&\leq (c_O + 1) \sum_{l_z=l_{z-1}}^{\gamma} 2^{l_z} \ldots \sum_{l_{z-2}=l_{z-3}}^{\gamma} 2^{l_{z-2}} O(T^{4/\rho}\log T) \\
&= O(T^{(2+2D_{\text{rel}})/\rho}\log T).
\end{aligned}$$

Since $R_O(T) \leq \binom{D}{2D_{\text{rel}}}(2D_{\text{rel}})! R_{O,\boldsymbol{j}^*}(T)$, we have $R_O(T) = O(T^{(2+2D_{\text{rel}})/\rho} \log T)$.

D. Balancing the regret due to exploitations and explorations

From the results of the previous subsections we have with probability $1-\delta$, $R_I(T) = \tilde{O}(T^{\rho/(1+\rho)})$ and $R_O(T) = \tilde{O}(T^{(2+2D_{\text{rel}})/\rho})$. Since $R_I(T)$ is increasing in ρ and $R_O(T)$ is decreasing in ρ there is a unique ρ for which they are equal. This unique solution is

$$\rho = \frac{2 + 2D_{\text{rel}} + \sqrt{4D_{\text{rel}}^2 + 16D_{\text{rel}} + 12}}{4 + 2D_{\text{rel}} + \sqrt{4D_{\text{rel}}^2 + 16D_{\text{rel}} + 12}}.$$

II. Appendix to the Numerical Results in [1]

A. Datasets

The datasets we have used in [1] are available at http://medianetlab.ee.ucla.edu/JSTSPdatasets/.

Breast Cancer (BC) [2]: The dataset consists of features extracted from the images of fine needle aspirate (FNA) of breast mass, that gives information about the size, shape, uniformity, etc., of the cells. Each instance of the dataset contains 10 attributes: clump thickness, cell size uniformity, cell shape uniformity, marginal adhesion, single epi cell size, bare nuclei, bland chromatin, normal nucleoli, mitoses and class. The class attribute takes two values "malignant" or "benign". We take the other 9 attributes as the context of the instance and normalized them to lie in $[0, 1]$. This normalization is done in the following way: maximum and minimum context values in the dataset are found. Minimum context value is subtracted from all contexts, then the result is divided by the difference between the maximum and minimum values such that they lie in $[0, 1]$. The prediction action belongs to the set $\{benign, malignant\}$. Reward is 1 when the prediction is correct and 0 otherwise. 50000 instances are created by duplication of the data and are randomly sequenced. Analysis of the data shows that 69% of the instances are labeled as "benign" while the rest is "labeled" as malignant. Instances arrive to the learner sequentially in an online fashion. When an instance comes, the learning algorithm selects an action based on its context.

Network Intrusion (NI) [2]: The network intrusion dataset from UCI archive [2] consists of a series of TCP connection records, labeled either as normal connections or as attacks. The data consists of 42 features. The names of the features can be found in http://kdd.ics.uci.edu/databases/kddcup99/kddcup.names. The set of features that are used in numerical results in [1] is correspond to the features in columns $[1:6 \;\; 23:30]$ of the 42 dimensional dataset plus the feature corresponding to the label of the previous instance which is located at the 42nd column of the dataset. Taken features are normalized to lie

in $[0, 1]$. The prediction action belongs to the set $\{attack, noattack\}$. Reward is 1 when the prediction is correct and 0 otherwise.

Network Intrusion with All Features (NI-AF) [2]: Same as NI but all 41 features available in the dataset except the label are included in the context vector.

Webpage Recommendation (WR) [6]: This dataset contains webpage recommendations of Yahoo! Front Page which is an Internet news website. Each instance of this dataset consists of (i) IDs of the recommended items and their features, (ii) context vector of the user, and (iii) user click information. For a recommended webpage (item), reward is 1 if the user clicks on the item and 0 otherwise. The context vector for each user is generated by mapping a higher dimensional set of features of the user including features such as gender, age, purchase history, etc. to $[0, 1]^5$. The details of this mapping is given in [6]. We select 5 items and consider $T = 10000$ user arrivals.

B. Learning algorithms

Next we briefly summarize the algorithms considered in our evaluation:

RELEAF: Our algorithm whose pseudocode is given in Fig. 2 in [1] with control numbers $D_{i,t}$ divided by 5000 to reduce the number of explorations.[1]

RELEAF-ALL: Same as RELEAF except that reward of the selected action is observed in every time slot. This version is useful when the reward of the selected action can be observed with no cost.

RELEAF-FO: Same as RELEAF except that it observes the rewards of all actions instead of the reward of the selected action. We refer to this version of our algorithm as RELEAF with full observation (RELEAF-FO). This algorithm is used in datasets BC and NI, in which the actions are predictions, and when the label is received, estimated rewards of all predictions can be updated. Unless otherwise specified RELEAF, RELEAF-ALL and RELEAF-FO are run with input parameters $L = 1$, $\delta = 0.01$, $\gamma_{\text{rel}} = 1$, $\rho = 2 + 2\sqrt{2}$.

Contextual zooming (CZ) [7]: This algorithm adaptively creates balls over the joint action and context space, calculates an index for each ball based on the history of selections of that ball, and at each time slot selects an action according to the ball with the highest index that contains a current action-context pair. The calculation of index involves adding an uncertainty term on top of the sample mean rewards that depends on the number of times a specific ball is selected and the radius of the ball. For CZ, the

[1]The theoretical bounds are proven to hold for worst-case context vector arrivals and reward distributions. In practice, the relevance relation and the order of action rewards are identified correctly with much less explorations.

Algorithm	Base classifiers	Prior training	Online Learning	Active learning
AM [9]	required	no	no	no
Adaboost [10]	required	required	no	no
Online Adaboost [10], **Blum** [12]	required	required	yes	no
CZ [7], **Hybrid-ϵ** [8] , **LinUCB** [6]	not required	not required	yes	no
RELEAF	not required	not required	yes	yes

TABLE I

PROPERTIES OF RELEAF, ENSEMBLE LEARNING METHODS AND OTHER CONTEXTUAL BANDIT ALGORITHMS.

Lipschitz constant L transforms the radius of the ball into the uncertainty about the expected reward caused by the size of the ball. Unless otherwise specified CZ runs with $L = 0.5$.

Hybrid-ϵ [8]: This algorithm is the contextual version of ϵ-greedy, which forms context-dependent sample mean rewards for the actions by considering the history of observations and decisions for groups of contexts that are similar to each other.

LinUCB [6]: This algorithm computes an index for each action by assuming that the expected reward of an action is a linear combination of different types of contexts. The action with the highest index is selected at each time step.

Ensemble Learning Methods Average Majority (AM) [9], Adaboost [10], Online Adaboost [11] and Blum's Variant of Weighted Majority (Blum) [12]: The goal of ensemble learning is to create a strong (high accuracy) classifier by combining predictions of base classifiers. Hence all these methods require base classifiers (trained a priori) that produce predictions (or actions) based on the context vector.

AM simply follows the prediction of the majority of the classifiers and does not perform active learning. Adaboost is trained a priori with 1500 instances, whose labels are used to compute the weight vector. Its weight vector is fixed during the test phase (it is not learning online); hence no active learning is performed during the test phase. In contrast, Online Adaboost always receives the true label at the end of each time slot. It uses a time window of 1000 past observations to retrain its weight vector. Similar to Online Adaboost, Blum also learns its weight vector online. The key differences between our algorithm and the methods that we compare against are given in Table I.

C. Breast cancer simulations

In this section we compare the performance of RELEAF, RELEAF-ALL and RELEAF-FO with other learning methods described in Section II-B. For the ensemble learning methods, there are 6 logistic regression base classifiers, each trained with a different set of 10 instances.

Algorithm	Performance				
	error %	missed %	false %	number of label observations	active learning cost for $c_O = 1$
AM	8.22	17.20	4.09	0 (no online learning)	0
Adaboost	4.60	3,82	4.97	1500 (to train weights)	1500
Online Adaboost	4.68	4.07	4.95	all labels are observed	50000
Blum	11.18	27.12	3.86	all labels are observed	50000
CZ	3.15	4.24	2.89	all labels are observed	50000
Hybrid-ϵ	8.83	11.77	7.48	all labels are observed	50000
LinUCB	10.67	7.27	12.22	all labels are observed	50000
RELEAF	1.88	1.93	1.86	2630	2630
RELEAF-ALL	1.24	1.19	1.36	all labels are observed	50000
RELEAF-FO	1.68	1.34	1.82	2630	2630

TABLE II

COMPARISON OF RELEAF WITH ENSEMBLE LEARNING METHODS AND OTHER CONTEXTUAL BANDIT ALGORITHMS FOR THE BREAST CANCER DATASET.

The simulation results are given in Table II. Since RELEAF-FO updates the reward of both predictions after the label is received, it achieves lower error rates compared to RELEAF. In this setting it is natural to assume that the reward of both predictions are updated, because observing the label gives information about which prediction is correct. RELEAF-ALL which observes all the labels has the lowest error rate.

Among the ensemble learning schemes Adaboost and Online Adaboost performs the best, however, their error rates are more than two times higher than the error rate of RELEAF and about three times higher than the error rate of RELEAF-FO. Although the number of actively obtained labels (explorations) for RELEAF and RELEAF-FO are higher than the initial training samples used to train Adaboost; neither RELEAF nor RELEAF-FO has a predetermined exploration size as Adaboost. This is especially beneficial when time horizon of interest is unknown or prediction performance is desired to be uniformly good over all time instances. CZ is the best among the other multi-armed bandit algorithms with 3.15% error, but worse than RELEAF which has 1.88% error.

D. Network intrusion simulations (15 dimensional context vector)

In this section we compare the performance of RELEAF, RELEAF-ALL and RELEAF-FO with other learning methods described in Section II-B. For the ensemble learning methods, the base classifiers are logistic regression classifiers, each trained with 5000 different instances from the NI. Comparison of

performances in terms of the error rate is given in Table III. We see that RELEAF-FO has the lowest error rate at 0.68%, more than two times better than any of the ensemble learning methods. All the ensemble learning methods we compare against use classifiers to make predictions, and these classifiers require a priori training. In contrast, RELEAF and RELEAF-FO do not require any a priori training, learn online and require only a small number of label observations (i.e. they can perform active learning).

CZ performs very poorly in this simulation because its learning rate is sensitive to Lipschitz constant that is given as an input to the algorithm which we set equal to 0.5 (the same values is used in all simulations). LinUCB performs the best in terms of the overall rate of error, but if we consider the error rate of RELEAF in exploitations it is better than LinUCB. This highlights the finding of Theorem 1 in [1] regarding RELEAF, which states that highly suboptimal actions are not chosen in exploitations with a high probability.

Algorithm	error %	exploitation error %	number of label observations
AM	3.07	N/A	0
Adaboost	3.1	N/A	1500
Online Adaboost	2.25	N/A	all
Blum	1.64	N/A	all
CZ	53	N/A	all
Hybrid-ϵ	8.8	N/A	all
LinUCB	0.27	N/A	all
RELEAF	1.19	0.24	398
RELEAF-ALL	1.07	0.22	all
RELEAF-FO	0.68	0.24	229

TABLE III

COMPARISON OF THE ERROR RATES OF RELEAF-FO WITH ENSEMBLE LEARNING METHODS FOR NETWORK INTRUSION DATASET.

E. Network intrusion simulations (41 dimensional context vector)

In this section we compare the performance of RELEAF and CZ for different L parameter values for the NI-AF dataset.

Table IV compares the performance of RELEAF and CZ as a function of the input parameter L. We see that RELEAF performs significantly better than CZ. Our numerical results illustrate that the error

percentage of RELEAF is decreasing in the L value, while the error percentage of RELEAF in exploitation slots (calculated only over the time slots in which RELEAF exploits) is increasing in the L value. This result is consistent with the observation that the number of explorations of RELEAF decreases with L. Error percentage in exploitation slots is increasing because the number of exploitation slots increases with L, while the accuracy of the sample mean estimates formed in exploration slots decreases with L. However, the error percentage (over all time slots) decreases. This is because the decrease in the error percentage due to exploiting more is larger than the increase in the error percentage due to exploiting with more inaccurate sample mean reward estimates. We see that the lowest error percentage for CZ is achieved when $L = 1$. for this dataset.

L	CZ Error %	RELEAF Error %	RELEAF Missed %	RELEAF False %	RELEAF Exploit error %	RELEAF Exploit %
0.1	24.22	17.56	10.74	23.57	0	58.70
0.5	61.68	5.30	2.53	7.74	0.34	89.12
1	23.89	3.92	2.07	5.55	0.43	92.33

TABLE IV

PERFORMANCE OF RELEAF AND CZ AS A FUNCTION OF INPUT PARAMETER L FOR THE NI-AF DATASET.

F. Webpage recommendation simulations

In this dataset only the click behavior of the user for the recommended item is observed. Moreover, it is reasonable to assume that the click behavior feedback is always available (no costly observations). The ensemble learning methods require availability of experts recommending actions and full reward feedback including the rewards of the actions that are not selected, to update the weights of the experts, hence they are not suitable for this dataset. In contrast, multi-armed bandit methods are more suitable since only the feedback about the reward of the chosen action is required. Hence we only compare RELEAF-ALL, CZ, LinUCB and Hybrid-ϵ for this dataset. We compare the click through rates (CTRs), i.e., average number of times the recommended item is clicked, of all algorithms in Table V. We observe that RELEAF-ALL has the highest CTR.

G. Identifying the relevant types

When RELEAF exploits at time t, it identifies a relevant type $\hat{c}_t(a)$ for every action $a \in \mathcal{A}$ and selects the arm with the highest sample mean reward according to its estimated relevant type. Hence, the value

Abbreviation	CTR
CZ	3.79
Hybrid-ϵ	6.41
LinUCB	6.06
RELEAF-ALL	6.62

TABLE V

COMPARISON OF THE CLICK THROUGH RATES (CTRS) OF RELEAF, CZ, HYBRID-ϵ AND LINUCB FOR WEBPAGE RECOMMENDATION DATASET.

of the context of the relevant type plays an important role on how well RELEAF performs.

For each dataset we choose a single action and for each chosen action show in Table VI the percentage of times a type is selected as the type that is relevant to that action in the time slots that RELEAF exploits. Since there are many types, only the 4 of the types which are selected as the relevant type for the corresponding action highest number of times are shown. For instance, for BC in 70% of the exploitation slots the type identified as the type relevant to action "predict benign" comes from a 3 element subset of the set of 9 types in the data. Similarly for NI the type identified as the type relevant to action "predict attack" comes from a 2 element subset of the set of 15 types in the data for 85% of the exploitation slots.

This information provided by RELEAF can be used to identify the relevance relation that is present in a dataset. For instance, consider the NI dataset. Since the type that is assigned as the estimated relevant type most of the times is only assigned in 45% of the exploitation slots, for the NI dataset we should have $D_{\text{rel}} > 1$. However, since the pair of types that are assigned as the estimated relevant type most of the times is assigned in 85% of the exploitation slots, we can conclude that approximately $D_{\text{rel}} \leq 2$ for the NI dataset.

REFERENCES

[1] C. Tekin and M. van der Schaar, "RELEAF: An algorithm for learning and exploiting relevance," *submitted to IEEE JSTSP*, 2014.

[2] K. Bache and M. Lichman, "UCI machine learning repository," http://archive.ics.uci.edu/ml, University of California, Irvine, School of Information and Computer Sciences, 2013.

[3] J. Gao, W. Fan, and J. Han, "On appropriate assumptions to mine data streams: Analysis and practice," in *Proc. IEEE ICDM*, 2007, pp. 143–152.

[4] M. M. Masud, J. Gao, L. Khan, J. Han, and B. Thuraisingham, "Integrating novel class detection with classification for concept-drifting data streams," in *Proc. ECML PKDD*, 2009, pp. 79–94.

Dataset	Action	highest rates of relevance			
		highest type-rate	2nd highest type-rate	3rd highest type-rate	4th highest type-rate
BC	predict "benign"	3-27%	1-22%	7-21%	2-12%
NI	predict "attack"	1-45%	15-40%	2-7%	4-5%
WR	recommend webpage a	3-46%	1-44%	2-8%	4-1%
WR	recommend webpage b	2-57%	1-32%	5-9%	4-1%

TABLE VI

AVERAGE NUMBER OF TIMES RELEAF IDENTIFIED A TYPE AS THE TYPE RELEVANT TO THE SPECIFIED ACTION IN EXPLOITATIONS.

[5] L. L. Minku and Y. Xin, "DDD: A new ensemble approach for dealing with concept drift," vol. 24, no. 4, pp. 619–633, 2012.

[6] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *Proceedings of the 19th International Conference on World Wide Web*. ACM, 2010, pp. 661–670.

[7] A. Slivkins, "Contextual bandits with similarity information," in *24th Annual Conference On Learning Theory*, 2011.

[8] D. Bouneffouf, A. Bouzeghoub, and A. L. Gançarski, "Hybrid-ε-greedy for mobile context-aware recommender system," in *Advances in Knowledge Discovery and Data Mining*. Springer, 2012, pp. 468–479.

[9] J. Gao, W. Fan, and J. Han, "On appropriate assumptions to mine data streams: Analysis and practice," in *Seventh IEEE International Conference on Data Mining (ICDM)*, 2007, pp. 143–152.

[10] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *Computational Learning Theory*. Springer, 1995, pp. 23–37.

[11] W. Fan, S. J. Stolfo, and J. Zhang, "The application of adaboost for distributed, scalable and on-line learning," in *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, 1999, pp. 362–366.

[12] A. Blum, "Empirical support for winnow and weighted-majority algorithms: Results on a calendar scheduling domain," *Machine Learning*, vol. 26, no. 1, pp. 5–23, 1997.