Discovering Action-Dependent Relevance : Learning from Logged Data

Onur Atan*, Cem Tekin†, Jie Xu‡, Mihaela van der Schaar*

*University of California Los Angeles, oatan@ucla.edu, mihaela@ee.ucla.edu

†Bilkent University, cemtekin@ee.bilkent.edu.tr

†University of Miami, jiexu@miami.edu

Abstract

In many learning problems, the decision maker is provided with various (types of) context information that she might utilize to select actions in order to maximize performance/rewards. But not all information is equally relevant: some context information may be more relevant to the decision problem at hand. Discovering and exploiting the most relevant context information speeds up learning, reduces costs and eliminates noise introduced by irrelevant context information. In many settings, discovering and exploiting the most relevant context information converts intractable problems into tractable problems. This paper develops methods to discover the relevant context information and learn the best actions to take on the basis of a logged bandit dataset and establishes performance bounds for these methods. These methods deal effectively with the two central challenges. The first is that only the rewards of actions actually taken will be observed; counterfactual reward observations are not available. The second is that the relevant context information can be different for different actions. Applications of these methods include clinical decision support systems, smart cities, recommender systems.

I. Introduction

As the world becomes ever more connected and instrumented, decision-makers have ever more rapid access to larger and larger datasets. Unfortunately, if every aspect of the data is necessary to make optimal or near-optimal decisions, the decision-maker's problem will generally be completely intractable. The work described here shows that if as is typically the case, only *some* of the information is relevant for each decision, then the decision-maker's problem can be made tractable. What the decision-maker must do is to discover what information is *relevant* in

the available data and use the relevant information to make good decisions. This paper presents a systematic framework and associated algorithms that enable a decision-maker to do this. The problem is complicated because different pieces of information may be relevant for different decisions. We derive performance guarantees for these algorithms. This work has numerous applications, including healthcare informatics, clinical decision support systems, recommendation systems, smart cities, etc.

Numerous practical applications can benefit from the algorithms developed here. One application is clinical decision support, in which the number of context types available for a given patient is often enormous: age, demographics, weight, test results, medical history etc. If all these context types were relevant for all actions, finding optimal or near-optimal treatment(s) would be a completely intractable task for (at least) two reasons: The first is that the amount of data required to learn effectively would be enormous – exponential in both the number of contexts and the number of potential treatments. The second is that, even if the data were available, analyzing it would be intractable – or at least impracticably slow. Fortunately, in most settings, the number of context types that are relevant for each given action is relatively small, and by learning the relevant context types, the problem of determining which medical test(s) should be administered and which treatment(s) should be applied can be solved tractably with manageably small datasets.

Several challenges need to be addressed in order to discover the relevant context types and learn the optimal actions from logged data in the most efficient way. Firstly, logged data only contains bandit feedback, which means that only the rewards of actions that have been taken are recorded. For instance, a patient was given treatment A but not treatment B and hence, only the effectiveness of treatment A on this patient is known and recorded in the dataset. Secondly, the learner can not control logging strategies on which the dataset is collected. In fact, the dataset can be fusion of different datasets collected based on different logging strategies. The problem is further complicated because the relevant context types may be action-dependent, meaning that for different actions the relevant context types can be different (see Figure **??**). For instance, while the effectiveness of one medical treatment may depend largely on the age of the patients, the effectiveness of another treatment may depend largely on the weight of the patients. These challenges make the existing approaches, such as feature selection methods [1], [2], unsuitable for the problem considered here of discovering the relevant context types that inform the various

actions. (For a comprehensive comparison, see section on related work below.)

In this paper, we develop a novel relevance test that relies on statistical estimation techniques to learn the relevant context types and the optimal action for each context vector. Our main contributions can be summarized as follows:

- We develop two algorithms that discover *action-dependent relevance relations* under different statistical assumptions. The algorithms use sample mean based estimators and are easy to implement.
- We derive upper bounds on the error probability and lower bounds on the sample complexity for both algorithms.
- We illustrate the superiority in terms of predictive accuracy of the proposed algorithms with respect to state-of-the-art methods using numerical results performed on a medical dataset.

II. Related Work

This paper relates to several strands of literature. Our work aims to learn the best actions to take by utilizing a logged dataset. Existing approaches with a similar goal can be organized into two categories. The first category uses propensity scoring [3], [4], [5], [6]. For instance, [6] studies counterfactual risk minimization by estimating propensity scores using knowledge of the logging strategy that is used to collect the data. In contrast, our methods apply to any logging strategy. The second category formalizes a supervised learning problem [7], [8] in which a regressor is trained for each action on data instances for which the reward of that action is observed. However, the relevance relation between action and contexts is not explicitly discovered. In contrast, we study problems where the rewards of actions only depend on a subset of relevant context types which are unknown and need to be discovered. By learning the relevant context types, the data sample size required to achieve the same level of confidence only depends on the number of relevant context types which is much smaller than the total context dimensionality in many practical applications.

The literature on relevance learning mostly consists of feature selection methods for classification problems. These methods can be divided into three categories – filter models, wrapper models, and embedded models [9]. Our method is most similar to filter models in which features are ranked based on certain criteria (e.g. Fisher score [10], mutual information based scores [11], [1], [2] and ReliefF and its variants [12], [13], and the features with the highest ranks are labeled

as relevant. However, these existing methods are developed for classification problems and they cannot easily handle datasets in which different actions will depend on different context types and the results of counterfactual actions are unknown. Moreover, while existing methods focus on classification problems, this work focuses on learning the optimal action to take.

We consider a setting in which the rewards of counterfactual actions are not observed and are also missing in the logged dataset. This relates to the literature on multi-armed bandits [14], [15], [16], [17] where the goal is to determine the select actions online to minimize the (learning) regret. While related, these latter two types of works study online learning problems. In contrast, our work focuses on learning based on a logged dataset in which the learner cannot control what features (contexts) to observe or what actions to select. We focus on this setting since in many applications, including medical informatics, only logged datasets are available to the learner.

III. Problem Formulation

Let $\mathcal{X}$ denote the space of D-dimensional context vectors and $\boldsymbol{x} = (x_1, x_2, \ldots, x_D)$ be an element in that space, with $x_d \in \mathcal{X}_d$ being a type-d context. Let $\mathcal{A} = \{1, 2, \ldots, K\}$ denote the set of actions with K being the total number of actions. An instance of the problem is specified by an unknown distribution $\mathbb{D}$ over tuples $(\boldsymbol{x}, \boldsymbol{y})$, where $\boldsymbol{x} \in \mathcal{X}$ is the context vector and $\boldsymbol{y} = (y_a)_{a\in\mathcal{A}} \in \mathcal{Y}^K$ is the vector of rewards where $\mathcal{Y}$ is the reward space and y_a is the reward associated with action a. We are given a dataset which consists of instances: a context vector $\boldsymbol{x}$, an action $a \in \mathcal{A}$ that was chosen (possibly as a function of $\boldsymbol{x}$ and historical information) and the reward y_a that resulted from applying this action in this context. Note that we do not assume knowledge of the action selection policy based on which this dataset was collected. For instance, in medical applications, the context $\boldsymbol{x}$ might contain patient information such as age, family history, results of medical tests, the action a might be the treatment or procedure selected for the patient and the reward y_a might represent the effectiveness of this treatment a. The dataset is

$$\mathcal{Z} = \big(\boldsymbol{x}(n), a(n), y_{a(n)}(n)\big)_{n=1}^{N}.$$

Our goal is to discover, from dataset $\mathcal{Z}$, the context types that were/are relevant for each action a and determine the optimal action for each relevant context or vector of contexts. It is important to note that the context(s) that are relevant for action a might be quite different

from the context(s) that are relevant for action a'. Note that only the reward associated with the selected action is available in the dataset – and not the rewards associated with actions that were not selected.

Let $\tilde{\boldsymbol{x}}$ denote an arbitatry sub-vector of $\boldsymbol{x}$ (containing some or all context types in $\boldsymbol{x}$). In general, the rewards of action a depend only on a subset of context types $\mathcal{R}(a) \subseteq \mathcal{D} = \{1, 2, \ldots, D\}$, which we call the *relevant types* of action a. The precise definition is:

Definition 1. *The set of information types $\mathcal{R}(a)$ is relevant to action a if $\mathcal{R}(a)$ is the minimal set satisfying the following statement : for all $\boldsymbol{x}_{\mathcal{R}(a)}$ and $\tilde{\boldsymbol{x}}$, $\mathbb{E}\left[y_a|\boldsymbol{x}_{\mathcal{R}(a)}, \tilde{\boldsymbol{x}}\right] = \mathbb{E}\left[y_a|\boldsymbol{x}_{\mathcal{R}(a)}\right]$.*

Let $\mathcal{R} = (\mathcal{R}(a))_{a\in\mathcal{A}}$ be the relevant context types and $\mathcal{IR}(a) = \mathcal{D} \setminus \mathcal{R}(a)$ be the remaining (irrelevant) context types of action a. It is important to keep in mind that the relevant types of actions are unknown *a priori* and must be discovered/learned. Given a complete vector of contexts $\boldsymbol{x}$ and a set of types $\mathcal{S} \subseteq \mathcal{D}$, we write $\boldsymbol{x}_{\mathcal{S}}$ for the restriction of $\boldsymbol{x}$ to $\mathcal{S}$ (i.e.; a context vector containing components in $\mathcal{S}$).

Figure **??** shows an illustration of such a relevance structure in which there are two context types $\mathcal{R} = \{1, 2\}$, two actions A and B, and $\mathcal{R}(\mathrm{A}) = \{1, 2\}$, $\mathcal{R}(\mathrm{B}) = \{2\}$. Because relevant contexts may be different for different actions, this is quite different from feature selection.

Let $\bar{y}_a^{\mathcal{S}}(\boldsymbol{x}_{\mathcal{S}})$ be the marginal expected reward of action a when the context information contains $\boldsymbol{x}_{\mathcal{S}}$, i.e., $\bar{y}_a^{\mathcal{S}}(\boldsymbol{x}_{\mathcal{S}}) = \mathbb{E}(y_a|\boldsymbol{x}_{\mathcal{S}})$ and similarly $\bar{y}_a$ be the marginal expected reward of action a, i.e., $\bar{y}_a = \mathbb{E}(y_a)$.

For any $\boldsymbol{x}_{\mathcal{R}}$, the optimal action is given by,

$$a^*(\boldsymbol{x}_{\mathcal{R}}) \triangleq \arg\max_a \bar{y}_a^{\mathcal{R}}(\boldsymbol{x}_{\mathcal{R}}).$$

Basically, $a^*(\boldsymbol{x}_{\mathcal{R}})$ is the action with the highest expected reward for the instances whose context vector with respect to the relevant types is $\boldsymbol{x}_{\mathcal{R}}$.

IV. Algorithm

A. Assumptions

Assumption 1. ***(General Assumptions)***

A1. *The information space is finite, i.e.* $|\mathcal{X}_d| \leq M < \infty, \forall d \in \mathcal{D}$.

A2. $\mathcal{Y} = [0, 1]$.

Assumption (A1) is not restrictive since any bounded context space can be discretized by partitioning, using techniques similar to the ones in [15]. Assumption (A2) is a standard one in the multi-armed bandit literature. Assuming that rewards are bounded (above and below), the specific form of $\mathcal{Y}$ is just a normalization. In addition to the assumptions above, we make statistical assumptions that will be used for the performance analysis of our proposed algorithms in two different scenarios.

Assumption 2. ***(Statistical Assumptions)***

A3. *The context types are statistically independent of each other, i.e.,* $\mathbb{P}(x_d|x_{\tilde{d}}) = \mathbb{P}(x_d)$ *for all* $x_d \in \mathcal{X}_d$ *and* $x_{\tilde{d}} \in \mathcal{X}_{\tilde{d}}$ *and* $d, \tilde{d} \in \mathcal{D}$.

A4. *No context type depends deterministically on any other context type; i.e.,* $0 < \mathbb{P}(\boldsymbol{x}_{\mathcal{S}}|\boldsymbol{x}_{\mathcal{S}'}) < 1$ *for all* $\boldsymbol{x}_{\mathcal{S}}$ *and* $\boldsymbol{x}_{\mathcal{S}'}$ *and* $\mathcal{S}, \mathcal{S}'$

We define a class of learning algorithms named Learn and Exploit the Action-dependent Relevance (LEAR). For scenarios in which (A3) holds, we develop an algorithm called LEAR-ICT, which associates relevance metric for each action and context type pair. For scenarios in which (A4) holds, we develop an algorithm called LEAR-CCT, which associates relevance metric for each action and context type tuples. LEAR-CCT is more general in the the sense that it works under a weaker assumption, but as we will show, it is much slower; i.e., requires many more samples than LEAR-ICT in order to achieve the same level of confidence in action reward estimation. A detailed comparison of these approaches is given in Section V-D and VI.

B. LEAR-ICT

This section proposes and analyzes an algorithm that discovers the relevant context type for each action from the logged dataset in the scenario in which each type of context is drawn from a distribution independently from the other types of contexts.

Given a set $\mathcal{S} \subseteq \mathcal{D}$, a context vector $\boldsymbol{x}_{\mathcal{S}}$ and an action a, let $\mathcal{Z}_a^{\mathcal{S}}(\boldsymbol{x}_{\mathcal{S}})$ denote the set of training instances n such that $\boldsymbol{x}_{\mathcal{S}}(n) = \boldsymbol{x}_{\mathcal{S}}$ and $a(n) = a$, and let $\mathcal{Z}_a$ denote the (larger) set of training instances n such that $a(n) = a$. The cardinality of these sets are denoted by $N_a^{\mathcal{S}}(\boldsymbol{x}_{\mathcal{S}}) = |\mathcal{Z}_a^{\mathcal{S}}(\boldsymbol{x}_{\mathcal{S}})|$ and $N_a = |\mathcal{Z}_a|$. Define the marginal sample mean estimator for the reward of action a on the

instances with context $\boldsymbol{x}_{\mathcal{S}}$ to be:

$$\hat{y}_a^{\mathcal{S}}(\boldsymbol{x}_{\mathcal{S}}) \triangleq \frac{1}{N_a^{\mathcal{S}}(\boldsymbol{x}_{\mathcal{S}})} \sum_{n \in \mathcal{Z}_a^{\mathcal{S}}(\boldsymbol{x}_{\mathcal{S}})} y(n).$$

Similarly, define the sample mean estimator for the reward of action a on all instances as

$$\hat{y}_a \triangleq \frac{1}{N_a} \sum_{n \in \mathcal{Z}_a} y(n).$$

Next, we summarize the steps of LEAR-ICT.

Step 1 : Using the reward estimates for all a, d, x, LEAR-ICT computes a relevance metric $h_d(a)$ for each context type - action pair (d, a):

$$h_d(a) \triangleq \sum_{x_d \in \mathcal{X}_d} \frac{N_a^d(x_d)}{N_a} |\hat{y}_a^d(x_d) - \hat{y}_a| \tag{1}$$

In view of the definitions of reward estimates, this relevance metric measures the weighted difference of the reward estimates when conditioning on the context type d and not conditioning. For example, if age is relevant to treatment a, then the survival rate (reward) of treatment a on a specific age group will be distinct from the survival rate of treatment a on the whole population. The relevance metric in (1) measures this difference.

Step 2 : Using the relevance metric for each pair (d, a), we can discover the relevant context types for each action a. Suppose that the R most relevant context types are to be discovered, then the R types which have the highest value of the relevance metric $h_d(a)$ are declared to be relevant. Let $\hat{\mathcal{R}}(a)$ be the relevant types for action a and $\hat{\mathcal{R}} = \left(\hat{\mathcal{R}}(a)\right)_{a \in \mathcal{A}}$ the set of relevant types for all actions.

Step 3 : The optimal action with respect to relevant types $\hat{\mathcal{R}}$ is determined as follows:

$$\hat{a}(\boldsymbol{x}_{\hat{\mathcal{R}}}) = \arg\max_a \hat{Y}_a^{\hat{\mathcal{R}}}(\boldsymbol{x}_{\hat{\mathcal{R}}}) \tag{2}$$

The pseudo-code for LEAR-ICT is given in the supplementary material.

C. LEAR-CCT

In this subsection, we develop a modified version of LEAR-ICT which works under the weaker independence assumption (A4). The difference in the algorithms is driven by the need to take account of the possibility that the combined effect of a large group of context types on the expected reward of action a is large even though the effect of each individual context type is

small. The basic steps in the two algorithms are similar but LEAR-CCT uses a different relevance metric for each action-type set pair $(a, \mathcal{S})$ as follows,

$$h_{\mathcal{S}}(a) \triangleq \max_{\boldsymbol{x}_{\mathcal{S}}, \boldsymbol{x'}_{\mathcal{S}}} \left(\hat{y}_a^{\mathcal{S}}(\boldsymbol{x}_{\mathcal{S}}) - \hat{y}_a^{\mathcal{S}}(\boldsymbol{x'}_{\mathcal{S}}) \right). \tag{3}$$

Then, using (3), the most R relevant context types associated with action a can be found as

$$\hat{\mathcal{R}}(a) = \arg\max_{\mathcal{S}:|\mathcal{S}|=R} h_{\mathcal{S}}(a). \tag{4}$$

Then, the optimal action can be found by using (2).

V. Performance Analysis

A. Error Metrics

In this subsection, we define our performance measures. We consider two types of errors. For the first type of error, the algorithm fails to identify the relevant context types. We call this the *relevance discovery* error and denote the probability of making this error for action a by $P_{\text{rel}}(a) \triangleq \mathbb{P}(\hat{\mathcal{R}}(a) \neq \mathcal{R}(a))$. For the second type of error, the algorithm fails to discover an action whose expected reward is in the ϵ-neighborhood of the optimal action for some $\epsilon > 0$. We call this *total error* and denote the probability of making this error for the instances with context containing $\boldsymbol{x}_{\mathcal{R}}$ as

$$P_{\text{err}}^{\epsilon}(\boldsymbol{x}_{\mathcal{R}}) \triangleq \mathbb{P}(\bar{y}_{\hat{a}(\boldsymbol{x}_{\hat{\mathcal{R}}})}(\boldsymbol{x}_{\mathcal{R}}) < \bar{y}_{a^*(\boldsymbol{x}_{\mathcal{R}})}(\boldsymbol{x}_{\mathcal{R}}) - \epsilon).$$

In the next subsection, we provide upper bounds on both the relevance discovery error and the total error made by our algorithms.

B. Theoretical Analysis of LEAR-ICT

First, we introduce an important notion that measures the inherent relevance of a context type to the decision problem. Let $\Delta_a^d(x_d) \triangleq |\bar{y}_a^d(x_d) - \bar{y}_a|$ be the relevance gap, which measures the expected reward difference created by the context x_d.

Proposition 1. *Under (A1), (A2) and (A3), for every $d \in \mathcal{IR}(a)$ and $x_d \in \mathcal{X}_d$, we have $\Delta_a^d(x_d) = 0$.*

Proposition 1 shows that irrelevant information types of action a do not matter for the expected reward of action a. Therefore, for any $d \in \mathcal{D}$ and action a, let

$$\Delta^d_{\text{ICT}}(a) \triangleq \sum_{x_d \in \mathcal{X}_d} \frac{N^d_a(x_d)\Delta^d_a(x_d)}{N_a} \tag{5}$$

be the relevance distance of action a created by context type d in the dataset $\mathcal{Z}$. Note that the distance $\Delta^d_{\text{ICT}}(a) = 0$ for $d \in \mathcal{IR}(a)$. If this distance for $d \in \mathcal{R}(a)$ is larger, then LEAR-ICT is able to learn the relevant context type d for action a faster. Therefore, the error probability bounds depend heavily on this problem specific quantity.

Theorem 1. ***Relevance Error Bound for LEAR-ICT*** *For all $a \in \mathcal{A}$, under (A1), (A2) and (A3), the relevance discovery error of LEAR-ICT is bounded as follows*

$$P_{rel}(a) \leq 4 \sum_{d \in \mathcal{D}} \sum_{r \in \mathcal{R}(a)} \sum_{x_d \in \mathcal{X}_d} \exp(-\frac{1}{8}\Delta^r_{ICT}(a)^2 N^d_a(x_d))$$

Proof. (Sketch) We bound $P_{\text{rel}}(a)$ using the probability of the event $\{\cup_{r \in \mathcal{R}(a)} \text{Rel}(r, a)\}$ where $\text{Rel}(r, a) = \{\exists d \in \mathcal{IR}(a) : h_d(a) \geq h_r(a)\}$. By applying the union bound and some other tricks, a Chernoff bound can then be utilized to obtain the claimed result. The details of the proof can be found in the supplementary material. □

Theorem 2. ***Total Error Bound for LEAR-ICT*** *For all $\boldsymbol{x}_{\mathcal{R}}$ and $\epsilon > 0$, under (A1), (A2) and (A3), the total error probability of LEAR-ICT is bounded as follows*

$$P^{\epsilon}_{err}(\boldsymbol{x}_{\mathcal{R}}) \leq 2 \sum_{a \in \mathcal{A}} \exp(-0.5\epsilon^2 N^{\mathcal{R}(a)}_a(\boldsymbol{x}_{\mathcal{R}(a)}))$$
$$+ 4 \sum_{a \in \mathcal{A}} \sum_{d \in \mathcal{D}} \sum_{r \in \mathcal{R}(a)} \sum_{x_d \in \mathcal{X}_d} \exp(-\frac{1}{8}\Delta^r_{ICT}(a)^2 N^d_a(x_d))$$

Proof. (Sketch) Let $\text{IRR} = \{\exists a \in \mathcal{A} : \hat{\mathcal{R}}(a) \neq \mathcal{R}(a)\}$. Then, the error bound can be divided into two parts as follows,

$$P^{\epsilon}_{\text{err}}(\boldsymbol{x}_{\mathcal{R}}) = \mathbb{P}(\bar{y}_{\hat{a}(\boldsymbol{x}_{\mathcal{R}})}(\boldsymbol{x}_{\mathcal{R}}) < \bar{y}_{a^*(\boldsymbol{x}_{\mathcal{R}})}(\boldsymbol{x}_{\mathcal{R}}) - \epsilon) + \mathbb{P}(\text{IRR}) \tag{6}$$

where the first part on the right-hand side can be bounded using the Chernoff-Hoeffding bound. For the second part, we use union bound and Theorem 1 □

Note that the bound given in Theorem 2 consists of two parts. The first part is the error due to sub-optimal action selection even though the relevant types are correctly identified. The second part is the error due to failing to identify the relevant types correctly.

Let $\Delta_{\min}^{\text{ICT}} = \min_{a\in\mathcal{A}} \min_{d\in\mathcal{R}(a)} \Delta_{\text{ICT}}^{d}(a)$ and $N_{\min}^{\text{ICT}} = \min_{a\in\mathcal{A}} \min_{d\in\mathcal{D}} \min_{x_d\in\mathcal{X}} N_a^d(x_d)$.

Corollary 1. ***Sample Complexity Bound for LEAR-ICT*** *Suppose $R = 1$, under (A1), (A2) and (A3), for all $\boldsymbol{x}_{\mathcal{R}}$ and $\epsilon, \delta > 0$, if*

$$N_{\min}^{ICT} \geq \max\left(\frac{2}{\epsilon^2}\log\left(\frac{4}{\delta}\right),\ \frac{8}{(\Delta_{min}^{ICT})^2}\log\left(\frac{4KMD}{\delta}\right)\right)$$

then $P_{err}^{\epsilon}(\boldsymbol{x}_{\mathcal{R}}) \leq \delta$.

Corollary 1 provides an upper bound on the number of observations required for LEAR-ICT to be able to select an action within an ϵ neighborhood of the optimal action with high probability $1-\delta$. Observe that $N_{\min}$ scales linearly with the number of relevant context types R, the dimension of context space D, the size of the context space M and the size of the action space K. This is a significant improvement compared to contextual learning in multi-armed bandit algorithms which scale quadratically with D (see e.g. [15]).

C. Theoretical Analysis of LEAR-CCT

To bound the errors made by LEAR-CCT, we need to introduce a modified relevance gap. Let

$$\Delta_a(\mathcal{S}) \triangleq \max_{\boldsymbol{x}_{\mathcal{S}}, \boldsymbol{x}_{\mathcal{S}}^{'}} \bar{y}_a^{\mathcal{S}}(\boldsymbol{x}_{\mathcal{S}}) - \bar{y}_a^{\mathcal{S}}(\boldsymbol{x}_{\mathcal{S}}^{'})$$

be the relevance gap, which measures the maximum expected reward difference conditional on the context types $\mathcal{D}$.

Proposition 2. *under (A1), (A2) and (A4), for all actions a and sets of contexts $\mathcal{S} \neq \mathcal{R}(a)$, we have $\Delta_a(\mathcal{S}) < \Delta_a(\mathcal{R}(a))$.*

Proposition 2 shows that the maximum difference of the expected rewards of the actions can be achieved by the relevant context types of action a.

The error probability of LEAR-CCT depends on a different relevance distance, which is defined as

$$\Delta_{\text{CCT}}(a) \triangleq \Delta_a(\mathcal{R}(a)) - \max_{\mathcal{S}\subseteq\mathcal{D}:|\mathcal{S}|=R, \mathcal{S}\neq\mathcal{R}(a)} \Delta_a(\mathcal{S}).$$

According to Proposition 2, we have $\Delta_{\text{CCT}}(a) > 0$ for all actions a. The reason why the error bounds of LEAR-CCT depend on a different relevance distance is that LEAR-CCT has to perform

Algorithms	LEAR-ICT ($R=2$)	LEAR-CCT ($R=2$)	CFS	ACL	LoR	LiR	SVM
p_1	73.46%	64.03%	47.81%	20.13%	62.34%	58.16%	65.77%
p_2	88.43%	79.51%	63.91%	39.98%	81.78%	67.54%	82.86%

TABLE I: Comparison of the LEAR with benchmark algorithms in terms of percent agreement

a combinatoral search to be able to discover the joint effects of the context types on the rewards of the action a. Let $N_a^*(R) = \min_{\mathcal{S}\subseteq\mathcal{D}:|\mathcal{S}|=R} \min_{\boldsymbol{x}_\mathcal{S}} N_a^{\mathcal{S}}(\boldsymbol{x}_\mathcal{S})$.

Theorem 3. ***Relevance Error Bound for LEAR-CCT*** *For all $a \in \mathcal{A}$, under (A1), (A2) and (A4), the relevance discovery error of LEAR-CCT is bounded as follows*

$$P_{rel}(a) \leq 2\binom{D}{R} M^R \exp(-\frac{1}{8}\Delta_{CCT}(a)^2 N_a^*(R))$$

Proof. We bound this by the probability of the event $\text{Rel}(a)$, which is defined as $\text{Rel}(a) = \{\exists \mathcal{S} \subseteq \mathcal{D} : |\mathcal{S}| = R, \mathcal{S} \neq \mathcal{R}(a), h_\mathcal{S}(a) \geq h_{\mathcal{R}(a)}(a)\}$. Then, by using the union bound and the Chernoff-Hoeffding bound, the claimed bound can be obtained. □

Theorem 4. ***Total Error Bound for LEAR-CCT*** *For all $\boldsymbol{x}_\mathcal{R}$ and $\epsilon > 0$, under (A1), (A2) and (A4), the error probability of LEAR-CCT is bounded as follows*

$$P_{err}^{\epsilon}(\boldsymbol{x}_\mathcal{R}) \leq 2\sum_{a\in\mathcal{A}} \exp(-0.5\epsilon^2 N_a^{\mathcal{R}(a)}(\boldsymbol{x}_{\mathcal{R}(a)})) + 2\binom{D}{R} M^R \sum_{a\in\mathcal{A}} \exp\left(-\frac{1}{8}\Delta_{CCT}^2(a) N_a^*(R)\right).$$

Proof. (Sketch) The same decomposition given in (6) holds in this case. We use the probability of $\{\cup_{a\in\mathcal{A}}\text{Rel}(a)\}$to bound $\mathbb{P}(\text{IRR})$. □

For the case R=1, a simpler expression falls out. For convenience, let $\Delta_{\min}^{\text{CCT}} = \min_{a\in\mathcal{A}} \Delta_{\text{CCT}}(a)$ and $N_{\min}^{\text{CCT}}(R) = \min_a N_a^*(R)$.

Corollary 2. ***Sample Complexity Bound for LEAR-ICT*** *Suppose $R = 1$, under (A1), (A2) and (A4), for all $\boldsymbol{x}_\mathcal{R}$ and $\epsilon, \delta > 0$, if*

$$N_{\min}^{CCT}(1) \geq \max\left(\frac{2}{\epsilon^2}\log\left(\frac{4}{\delta}\right),\ \frac{8}{(\Delta_{\min}^{CCT})^2}\log\left(\frac{4KMD}{\delta}\right)\right)$$

then $P_{err}^{\epsilon}(\boldsymbol{x}_\mathcal{R}) \leq \delta$.

Number of training instances ($\times 10^3$)	2	4	10	20	30	40	50
LEAR-ICT ($R = 2$)	67.4%	73.4%	73.0%	73.3%	73.3%	73.3%	73.4%
LEAR-CCT ($R = 2$)	57.9%	64.0%	69.4%	71.6%	72.7%	74.5%	74.9%

TABLE II: Comparison of the LEAR-ICT and LEAR-CCT in terms of percent agreement

D. Comparison of LEAR-ICT and LEAR-CCT

In this subsection, we compare the two proposed algorithms. Since these two algorithms differ only in the selection of the relevant context types, we use Theorem 1 and 3 in the comparison.

- The discovery of relevant context types for LEAR-ICT heavily depends on the independence assumption. If this assumption does not hold in reality, LEAR-ICT may converge to a suboptimal relevance set $\mathcal{R}$ whereas LEAR-CCT converges to correct relevance set. This may lead to performance degredation for LEAR-ICT.
- Because LEAR-CCT performs a combinatorial search, it requires more samples than LEAR-ICT in order to correctly discover the relevance relation. In view of Theorem 3, the relevance error probability of LEAR-CCT scales with $\mathrm{O}(\binom{D}{R} M^R)$, while in view of Theorem 1 the relevance error probability of LEAR-ICT scales with $O(RDM)$. This will lead to a performance degradation for LEAR-CCT with respect to LEAR-ICT if $R >> 1$.

VI. Numerical Results

A. Dataset

Training and comparing LEAR-ICT and LEAR-CCT requires large dataset. We perform experiments on 50000 breast cancer patient cases characterized by 18 different contexts. These cases are generated by examining the characteristics of individuals studied in 80 clinical studies that study 6 different chemotherapy treatments. The reward of a treatment on specific patient is the success rate of the treatment on the population in which the patient belongs. (The details of the dataset are omitted here due to anonymity reasons, but a link to the dataset and the details will be added in the final version.)

B. Benchmarks

We compare the performance of our algorithms with Logistic Regression (LoR), Linear Regression (LiR), Sup- port Vector Machines (SVM), Correlation Feature Selection (CFS), a well-known feature selection algorithm [18] and All Contextual Learning (ACL), a contextual

learning algorithm which uses all features along with the rewards of the actions played in the past to make the decision, which is a modified offline version of the contextual bandit algorithm in [15].

We utilize a training set of 4,000 patients. The remaining patients provide the test set. Standard 50-fold stratified cross-validation was applied, and therefore, no training data were used during testing of the model, but 50 different models were used to derive the final test results.

C. Results

Comparison with Benchmarks : Given a patient, our algorithm or any of these other benchmark algorithms recommend a course of treatment corresponding to a reference clinical study that includes patients having the same characteristics (contexts) as the considered patient. As a performance metric, we take the fraction/percentage of "correct" treatment recommendations (i.e. the recommendations of the algorithms correspond to the best recommendations made in the clinical studies for that type of patient) to be the success rate for the algorithm in question. (Notice that the best course of treatment in the clinical study may not promise a good outcome: some cancers are not treatable.) We define the percent agreements (p_1 and p_2) as the fraction of times the first choice of treatment or the first and second choice of treatment predicted by the algorithm in question match the best recommendations made in the corresponding clinical study. Table I demonstrates that LEAR-ICT recommends the same treatment as that selected as the top choice by the reference clinical study for that patient 73.4% and as the top two best choices 88.4% . This is 7.7% better than the next best approach (i.e., SVMs) in terms of matching the top treatment, and 5.6% better in terms of matching the top two treatments recommended by the reference clinical study.

Comparison of LEAR-ICT and LEAR-CCT : Because a large academic medical center may be able to obtain a sufficient cohort to train treatment recommendation algorithms such as LEAR, it is often informative to know the amount of cases needed to ensure that the near-optimal treatment is recommended with a high probability. Our simulation results (Table II) show that: LEAR-ICT requires 4,000 cases and LEAR-CCT requires 40,000 cases to reach near-optimal action recommendation. After 35,000 cases, LEAR-CCT works better than LEAR-ICT because LEAR-CCT exploits the correlations existing among the context types and $35,000$ cases are enough to enable LEAR-CCT to dis- cover relevance.

VII. Conclusions

In this paper, we formalized the problem of discovering action-dependent context types and learning the best action to take based on the context information contained in the relevant set of types. We developed two simple algorithms based on different notions of relevance metric. These algorithms come with provable guarantees: We show upper bounds on the probability of error and lower bounds on the number of samples needed to be able to discover an action whose expected reward is in an ϵ-neighborhood of the optimal action with high probability. The proposed algorithms have a wide applicability.

References

[1] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," in *International Conference on Machine Learning (ICML)*, vol. 3, 2003, pp. 856–863.

[2] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 8, pp. 1226–1238, 2005.

[3] A. Strehl, J. Langford, L. Li, and S. M. Kakade, "Learning from logged implicit exploration data," in *Advances in Neural Information Processing Systems*, 2010, pp. 2217–2225.

[4] L. Li, R. Munos, and C. Szepesvári, "Toward minimax off-policy value estimation." in *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, 2015, pp. 608–616.

[5] J. Langford, L. Li, and M. Dudík, "Doubly robust policy evaluation and learning," in *Proceedings of the 28th International Conference on Machine Learning (ICML)*, 2011, pp. 1097–1104.

[6] A. Swaminathan and T. Joachims, "Counterfactual risk minimization: Learning from logged bandit feedback," in *Proceedings of the 32nd International Conference on Machine Learning*, 2015, pp. 814–823.

[7] A. Beygelzimer and J. Langford, "The offset tree for learning with partial labels," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pp. 129–138.

[8] B. Zadrozny, J. Langford, and N. Abe, "Cost-sensitive learning by cost-proportionate example weighting," in *Data Mining, ICDM. Third IEEE International Conference on*, 2003, pp. 435–442.

[9] J. Tang, S. Alelyani, and H. Liu, "Feature selection for classification: A review," *Data Classification: Algorithms and Applications. Editor: Charu Aggarwal, CRC Press In Chapman & Hall/CRC Data Mining and Knowledge Discovery Series*, 2014.

[10] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. John Wiley & Sons, 2012.

[11] D. Koller and M. Sahami, "Toward optimal feature selection," 1996.

[12] K. Kira and L. A. Rendell, "A practical approach to feature selection," in *Proceedings of the ninth international workshop on Machine learning*, 1992, pp. 249–256.

[13] M. Robnik-Šikonja and I. Kononenko, "Theoretical and empirical analysis of relieff and rrelieff," *Machine learning*, vol. 53, no. 1-2, pp. 23–69, 2003.

[14] T. Lu, D. Pál, and M. Pál, "Contextual multi-armed bandits," in *International Conference on Artificial Intelligence and Statistics*, 2010, pp. 485–492.

[15] A. Slivkins, "Contextual bandits with similarity information," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 2533–2568, 2011.

[16] J. Langford and T. Zhang, "The epoch-greedy algorithm for contextual multi-armed bandits," *Advances in Neural Information Processing Systems (NIPS)*, vol. 20, pp. 1096–1103, 2008.

[17] W. Chu, L. Li, L. Reyzin, and R. E. Schapire, "Contextual bandits with linear payoff functions," in *International Conference on Artificial Intelligence and Statistics*, 2011, pp. 208–214.

[18] M. A. Hall, "Correlation-based feature selection for machine learning," Ph.D. dissertation, The University of Waikato, 1999.