Unsupervised Sequential Sensor Acquisition

Anonymous Author 1
Unknown Institution 1

Anonymous Author 2
Unknown Institution 2

Anonymous Author 3
Unknown Institution 3

Abstract

standard setup,
duce it first .

a bit more com-
ostract will need
e settle on the re-

Sequential sensor acquisition problems (SAP) arise in many application domains including medical-diagnostics, security and surveillance. SAP architecture is organized as a cascaded network of "intelligent" sensors that produce decisions upon acquisition. Sensors must be acquired sequentially and comply with the architecture. Our task is to identify the sensor with optimal accuracy-cost tradeoff. We formulate SAP as a version of the stochastic partial monitoring problem with side information and *unusual* reward structure. Actions correspond to choice of sensor and the chosen sensor's parents decisions are available as side information. Nevertheless, what is atypical, is that we do not observe the reward/feedback, which a learner often uses to reject suboptimal actions. Unsurprisingly, with no further assumptions, we show that no learner can achieve sublinear regret. This negative result leads us to introduce the notion of weak dominance on cascade structures. Weak dominance supposes that a child node in the cascade has higher accuracy whenever its parent's predictions are correct. We then empirically verify this assumption on real datasets. We show that weak dominance is a maximal learnable set in the sense that we must suffer linear regret for any non-trivial expansion of this set. Furthermore, by reducing SAP to a special case of multi-armed bandit problem with side information we show that for any instance in the weakly dominant we only suffer a sublinear regret.

1 Introduction

Sequential sensor acquisition arises in many scenarios where we have a diverse collection of sensors with differing costs and accuracy. In these applications, to minimize costs, one often chooses inexpensive sensors first; and based on their outcomes, one sequentially decides whether or not to acquire more expensive sensors. For instance, in security systems(see [1] and other medically oriented examples), costs can arise due to sensor availability and delay. A suite of sensors/tests including inexpensive ones such as magnetometers, video feeds, to more expensive ones such as millimeter wave imagers are employed. These sensors are typically organized in a hierarchical architecture with low-cost sensors at the top of the hierarchy. The task is to determine which sensor acquisitions lead to maximizing accuracy for the available cost-budget.

These scenarios motivate us to propose the unsupervised sequential sensor acquisition problem (SAP). Our SAP architecture is organized as a cascaded network of intelligent sensors. The sensors when utilized to probe an instance, outputs a prediction of the underlying state of the instance (anomaly or normal, threat or no-threat etc.). Sensors are ordered with respect to increasing cost and accuracy. While the costs are assumed to be known a priori, the exact misclassification rate of a sensor is unknown. This setup is realistic in security and surveillance scenarios because sensors are often required to be deployed in new domains/environments with little or no opportunity for re-calibration.

We assume that the scenario is played over multiple rounds with an instance associated with each round. Sensors must be acquired sequentially and comply with the cascade architecture in each round. The learner's goal is to figure out the hidden, stochastic state of the instance based on the sensor outputs. Since the learner knows that the sensors are ordered from least to most accurate he/she can use the most accurate sensor among his/her acquired sensors for prediction. Nevertheless, since the learner does not know the sensor accuracy he/she faces the dilemma of as to which sensor to use for predicting this state.

Preliminary work. Under review by AISTATS 2017. Do not distribute.

We frame our problem as a version of stochastic partial monitoring problem [2] with *atypical* reward structure. As is common, we pose the problem in terms of competitive optimality. We consider a competitor who can choose an optimal action with the benefit of hindsight. Our goal is to minimize cummulative regret based on learning the optimal action based on observations that are observed during multiple rounds of play.

Stochastic partial monitoring problem is itself a generalization of multi-armed bandit problems, the latter going back to [3]. In our context, we view sensors choices as actions. The availability of predictions of parent sensors of a chosen sensor is viewed as side observation. Recall that in a stochastic partial monitoring problem a decision maker needs to choose the action with the lowest expected cost by repeatedly trying the actions and observing some feedback. The decision maker lacks the knowledge of some key information, such as in our case, the misclassification error rates of the classifiers, but had this information been available, the decision maker could calculate the expected costs of all the actions (sensor acquisitions) and could choose the best action (sensor). The feedback received by the decision maker in a given round depends stochastically on the unknown information and the action chosen. Bandit problems are a special case of partial monitoring, where the key missing information is the expected cost for each action (or arm), and the feedback is simply the noisy version of the expected cost of the action chosen. In the *unsupervised* version considered here and which we call the unsupervised *sequential sensor acquisition problem* (SAP), the learner only observes the outputs of the classifiers, but not the label to be predicted over multiple rounds in a stochastic, stationary environment.

This leads us to the following question: Can a learner still achieve the optimal balance in this case? We first show that, unsurprisingly, with no further assumptions, no learner can achieve sublinear regret. This negative result leads us to introduce the notion of weak dominance on tests. It is best described as a relaxed notion of strong dominance. Strong dominance states that a sensor's predictions are almost surely correct whenever the parent nodes in the cascade are correct. We empirically demonstrate that weak dominance appears to hold by evaluating it on several real datasets. We also show that in a sense weak dominance is fundamental, namely, without this condition there exist problem instances that result in linear regret. On the other hand whenever this condition is satisfied there exist polynomial time algorithms that lead to sublinear ($O(\sqrt{T})$) cummulative regret.

been defined?

nance has not been

Our proof of sublinear regret is based on reducing SAP to a version of multi-armed bandit problem (MAB) with side-observation. The latter problem has already been shown to have sub-linear regret in the literature. In our reduction, we identify sensor nodes in the cascade as the bandit arms. The payoff of an arm is given by loss from the corresponding stage, and the side observation structure is defined by the feedback graph induced by the cascade. We then formally show that there is a one-to-one mapping between algorithms for SAP and algorithms for MAB with side-observation. In particular, under weak dominance, the regret bounds for MAB with side-observation then imply corresponding regret bounds for SAP.

2 Related Work

In contrast to our SAP setup there exists a wide body of literature dealing with fully supervised sensor acquisition. Like us [4] [5] [6] also deal with cascade models. However, unlike us these works focus on prediction-time cost/accuracy tradeoffs. In particular they assume that a fully labeled training dataset is provided for test-time use. This dataset has sensor feature data, sensor decisions as well as annotated ground-truth labels. The goal for the learner is to learn a policy for acquiring sensors based on training data to optimize cost/accuracy during test-time. The work of [7] decide when to quit a cascade that leads to better decisions to maximize throughput against error rates. Full feedback about classification accuracy is assumed.

Active classification: [8] considers the problem of PAC learning the best "active classifier", a classifier that decides about what tests to take given the results of previous tests to minimize total cost when both tests and misclassification errors are priced. Unlike us they only consider the batch, supervised learning. The same setting is also studied under hard budget constraints in [9] and its applications in imaging and computer vision systems are explored in [10, 11]).

Cs: I suspe
than this: I
we had a se
"their mode
model of th
(this would
joint proba

Cs: Actuall
need to goo

M: Added a
erence, not
clude more

Online learning: In [12], the decision maker can opt to pay for additional observations of the costs associated with other arms. Unlike ours this setting is not unsupervised. In [13], online learning with costly features and labels is studied. In each round, learner has to decide which features to observe, where each feature costs some money. The learner can also decide not to observe the label, but the learner always has the option to observe the label. Again this setting is not unsupervised.

Partial monitoring: General theory of [2] applies to the so-called finite problems (unknown "key information") is an element of the probability simplex. [14] considers special case when the payoff is also observed (akin to the side-observation problem of [15][16],[17]).

M: Need to
there seems
on this in I
and NIPS'1

The paper is organized as follows: in Section 3 we give a brief background on online learning problems and discuss information structure in these setups. In Section 4 we introduce SAP as a general online learning problem where feedback reveals no information on loss/reward of actions. In Sectjon 5 we identify conditions under which optimal action can be learned in SAP. In Section 6 we establish that SAP is regret equivalent to a stochastic multi-armed bandits with side-observations when it satisfies strong dominance property. When this property holds, Section 7 gives an algorithm to solve SAP efficiently. We conclude in Section 9 with a discussion on further extensions.

3 Background

The purpose of this section is to present some necessary background material that will prove to be useful later. In particular, we introduce a number of sequential decision making problems, namely stochastic partial monitoring, bandits and bandits with side-observations, which we will build upon later.

First, a few words about our notation: We will use upper case letters to denote random variables. The set of real numbers is denoted by $\mathbb{R}$. For positive integer n, we let $[n] = \{1, \dots, n\}$. We let $M_1(\mathcal{X})$ to denote the set of probability distributions over some set $\mathcal{X}$. When $\mathcal{X}$ is finite with a cardinality of $d \doteq |\mathcal{X}|$, $M_1(\mathcal{X})$ can be identified with the d-dimensional probability simplex.

In a *stochastic partial monitoring problem* a learner interacts with a stochastic environment in a sequential manner. In round $t = 1, 2, \dots$ the learner chooses an action A_t from an action set $\mathcal{A}$, and receives a feedback $Y_t \in \mathcal{Y}$ from a distribution p which depends on the action chosen and also on the environment instance identified with a "parameter" $\theta \in \Theta$: $Y_t \sim p(\cdot; A_t, \theta)$. The learner also incurs a reward R_t, which is a function of the action chosen and the unknown parameter θ: $R_t = r(A_t, \theta)$. The reward may or may not be part of the feedback for round t. The learner's goal is to maximize its total expected reward. The family of distributions $(p(\cdot; a, \theta))_{a,\theta}$ and the family of rewards $(r(a, \theta))_{a,\theta}$ and the set of possible parameters Θ are known to the learner, who uses this knowledge to judiciously choose its next action to reduce its uncertainty about θ so that it is able to eventually converge on choosing only an optimal action $a^*(\theta)$, achieving the best possible reward per round, $r^*(\theta) = \max_{a \in \mathcal{A}} r(a, \theta)$. The quantification of the learning speed is given by the expected regret $\mathfrak{R}_n = nr^*(\theta) - \mathbb{E}\left[\sum_{t=1}^n R_t\right]$, which, for brevity and when it does not cause confusion, we will just call regret. A sublinear expected regret, i.e., $\mathfrak{R}_n/n \to 0$ as $n \to \infty$ means that the learner in the long run collects almost as much reward on expectation as if the optimal action was known to it. Such a learner is called Hannan consistent. In some cases it is more natural to define the problems in terms of costs as opposed to rewards; in such cases the definition of regret is modified appropriately. Transforming between costs and rewards is trivial by flipping the sign of the rewards and costs.

Cs: Not su
sistency is
dom averag
zero with p

A wide range of interesting sequential learning scenarios can be cast as partial monitoring. One special case is bandit problems when $\mathcal{Y}$ is the set of real numbers and $r(a, \theta)$ is the mean of distribution $p(\cdot; a, \theta)$: Thus, in a bandit problem in evert round the learner chooses an action A_t based on its past observations and receives the noisy reward $Y_t \sim p(\cdot; A_t, \theta)$ as feedback. A bandit problem is special in that the observation Y_t and the reward are directly tied. Another special case is finite-armed *bandits with side-observations* [15], where each action $a \in \mathcal{A}$ is associated with a neighbor-set $\mathcal{N}(a) \subset \mathcal{A}$ and the set of neighborhoods is known to the learner from the beginning. The learner upon choosing action $A_t \in \mathcal{A}$ receives noisy reward observations for each action in $\mathcal{N}(A_t)$: $Y_t = (Y_{t,a})_{a \in N(A_t)}$, where $Y_{t,a} \sim p_r(\cdot; a, \theta)$, and $\mathbb{E}[Y_{t,a}] = r(a, \theta)$. (The action chosen may or may not be an element of $N(A_t)$.) The reader can readily verify that this problem can also be cast as a partial monitoring problem by defining $\mathcal{Y}$ as the set $\cup_{i=0}^K \mathbb{R}^i$ and defining the family of distributions $(p(\cdot; a, \theta))_{a,\theta}$ such that $Y_t \sim p(\cdot; A_t, \theta)$. Finally, we note in passing that while we called Θ a parameter set, we have not equipped Θ with any structure. As such, the framework is able to model both bona fide parametric settings (e.g., Bernoulli rewards) and the so-called non-parametric settings. For example, K-armed bandits with reward distributions supported over $[0, 1]$ can be modelled by choosing Θ as the set of all K-tuples $\theta := (\theta_1, \dots, \theta_K)$ of distributions over $[0, 1]$ and setting $p(\cdot; a, \theta) = \theta_a(\cdot)$. More generally, we can identify Θ with set of instances $(p(\cdot; a, \theta), r(a, \theta))_{\theta \in \Theta}$. In what follows, when convenient, we will use this identification and will view elements of Θ as a pair p, r where $p(\cdot; a)$ is a probability distribution over $\mathcal{Y}$ for each $a \in \mathcal{A}$ and r is a map from $\mathcal{A}$ to the reals.

4 Unsupervised Sensor Acquisition Problem

Cs: I compressed the problem spec. We don't want the reader to get bored.

The formal problem specification of the unsupervised, stochastic, cascaded sensor acquisition problem is as follows: A problem instance is specified by a pair $\theta = (P, c)$, where P is a distribution over the $K + 1$ dimensional hypercube, and c is a K-dimensional, non-negative valued vector of costs. While c is known

Cs: I added
caded. Late
sider altern
useful to ha
distinguish
defined her
natives.

to the learner from the start, P is initially unknown. We henceforth identify problem instance θ by P only as c is assumed to fixed and known. The instance parameters specify the learner-environment interaction as follows: In each round for $t = 1, 2, \dots$, the environment generates a $K+1$-dimensional binary vector $Y = (Y_t, Y_t^1, \dots, Y_t^K)$ chosen at random from P. Here, Y_t^i is the output of sensor i, while Y_t is a (hidden) label to be guessed by the learner. Simultaneously, the learner chooses an index $I_t \in [K]$ and observes the sensor outputs $Y_t^1, \dots, Y_t^{I_t}$. The sensors are known to be ordered from least accurate to most accurate, i.e., $\gamma_k \doteq \mathbb{P}\left(Y_t \neq Y_t^k\right)$ is decreasing with k increasing. Knowing this, the learner's choice of I_t also indicates that he/she chooses I_t to predict the unknown label Y_t. Observing sensors is costly: The cost of choosing I_t is $C_{I_t} \doteq c_1 + \cdots + c_{I_t}$. The total cost suffered by the learner in round t is thus $C_{I_t} + \mathbb{I}\{Y_t \neq Y_t^{I_t}\}$. The goal of the learner is to compete with the best choice given the hindsight of the values $(\gamma_k)_k$. The expected regret of learner up to the end of round n is $\mathfrak{R}_n = (\sum_{t=1}^n \mathbb{E}\left[C_{I_t} + \mathbb{I}\{Y_t \neq Y_t^{I_t}\}\right]) - n \min_k (C_k + \gamma_k)$. For future reference, we let $c(k,\theta) = \mathbb{E}\left[C_k + \mathbb{I}\{Y_t \neq Y_t^k\}\right] (= C_k + \gamma_k)$ and $c^*(\theta) = \min_k c(k,\theta)$. Thus, $\mathfrak{R}_n = (\sum_{t=1}^n \mathbb{E}\left[c(I_t,\theta)\right]) - nc^*(\theta)$. In what follows, we shall denote by $\mathcal{A}^*(\theta)$ the set of optimal actions of θ and we let $a^*(\theta)$ denote the optimal action that has the smallest index. Thus, in particular, $a^*(\theta) = \min \mathcal{A}^*(\theta)$. Note that even if $i < j$ are optimal actions, there can be suboptimal actions in the interval $[i,j](= [i,j] \cap \mathbb{N})$ (e.g., $\gamma_1 = 0.3$, $C_1 = 0$, $\gamma_2 = 0.25$, $C_2 = 0.1$, $\gamma_3 = 0$, $C_3 = 0.3$. Next, for future reference note that one can express optimal actions from the viewpoint of marginal costs and marginal error. In particular an action i is optimal if for all $j > i$ the marginal increase in cost, $C_j - C_i$, is larger than the marginal decrease in error, $\gamma_i - \gamma_j$: $\forall\, j \geq i$

$$\underbrace{C_j - C_i}_{\text{Marginal Cost}} \geq \gamma_i - \gamma_j = \underbrace{E\left[\mathbb{I}\{Y_t \neq Y_t^i\} - \mathbb{I}\{Y_t \neq Y_t^j\}\right]}_{\text{Marginal Decrease in Error}}. \tag{1}$$

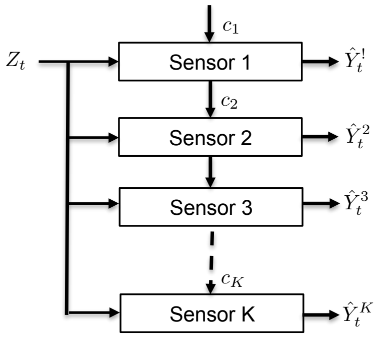

Figure 1: Structure of sensors in Sequential Sensor Acquisition. $Z_t = (X_t, Y_t)$, where X_t is feature vector and Y_t is its label (not observed), is sequentially input to sensors. After observing $\hat{Y}_t^t$, $\hat{Y}_t^{i+1}$ is observed by incurring additional cost of c_{i+1}

his figure and cap-

noving this to foot-

5 When is SAP Learnable?

Let Θ_{SA} be the set of all stochastic, cascaded sensor acquisition problems. Thus, $\theta \in \Theta_{\text{SA}}$ such that if $Y \sim \theta$ then $\gamma_k(\theta) := \mathbb{P}\left(Y \neq Y^k\right)$ is a decreasing sequence. Given a subset $\Theta \subset \Theta_{\text{SA}}$, we say that Θ is *learnable* if there exists a learning algorithm $\mathfrak{A}$ such that for any $\theta \in \Theta$, the expected regret $\mathbb{E}\left[\mathfrak{R}_n(\mathfrak{A},\theta)\right]$ of algorithm $\mathfrak{A}$ on instance θ is sublinear. A subset Θ is said to be a maximal learnable problem class if it is learnable and for any $\Theta' \subset \Theta_{\text{SA}}$ superset of Θ, Θ' is not learnable. In this section we study two special learnable problem classes, $\Theta_{\text{SD}} \subset \Theta_{\text{WD}}$, where the regularity properties of the instances in Θ_{SD} are more intuitive, while Θ_{WD} can be seen as a maximal extension of Θ_{SD}.

Let us start with some definitions. Given an instance $\theta \in \Theta_{\text{SA}}$, we can decompose θ (or P) into the joint distribution P_S of the sensor outputs $S = (Y^1, \dots, Y^k)$ and the conditional distribution of the state of the environment, given the sensor outputs, $P_{Y|S}$. Specifically, letting $(Y,S) \sim P$, for $s \in \{0,1\}^K$ and $y \in \{0,1\}$, $P_S(s) = \mathbb{P}\left(S = s\right)$ and $P_{Y|S}(y|s) = \mathbb{P}\left(Y = y | S = s\right)$. We denote this by $P = P_S \otimes P_{Y|S}$. A learner who observes the output of all sensors for long enough is able to identify P_S with arbitrary precision, while $P_{Y|S}$ remains hidden from the learner. This leads to the following statement:

Proposition 1. *A subset $\Theta \subset \Theta_{\text{SA}}$ is learnable if and only if there exists a map $a : M_1(\{0,1\}^K) \to [K]$ such that for any $\theta \in \Theta$ with decomposition $P = P_S \otimes P_{Y|S}$, $a(P_S)$ is an optimal action in θ.*

An action selection map $a : M_1(\{0,1\}^K) \to [K]$ is said to be *sound* for an instance $\theta \in \Theta_{\text{SA}}$ with $\theta = P_S \otimes P_{Y|S}$ if $a(P_S)$ selects an optimal action in θ. With this terminology, the previous proposition says that a set of instances Θ is learnable if and only if there exists a sound action selection map for all the instances in Θ.

A class of sensor acquisition problems that contains instances that satisfy the so-called *strong dominance* condition will be shown to be learnable:

Definition 1 (Strong Dominance)**.** *An instance $\theta \in \Theta_{\text{SA}}$ is said to satisfy the* strong dominance property *if it holds in the instance that if a sensor predicts correctly then all the sensors in the subsequent stages of the cascade also predict correctly, i.e., for any $i \in [K]$,*

$$Y^i = Y \;\Rightarrow\; Y^{i+1} = \cdots = Y^K = Y \tag{2}$$

almost surely (a.s.) where $(Y, Y^1, \dots, Y^K) \sim P$.

dataset	γ_1	γ_2	δ_{12}
diabetic	0.288	0.219	0.075
heart	0.305	0.169	0.051

Table 1: Error statistics

Before we develop this concept further we will motivate strong dominance based on experiments on a few real-world datasets. Table 5 lists the error probabilities of the classifiers (sensors) for the heart and diabetic datasets from UCI repository. For both the datasets, γ_1 denotes the test error of an SVM classifier (linear) trained with low cost features and γ_2 denotes test error of SVM classifier trained using both low and high-cost features (cf. Section 8). The last column lists $\delta_{12} := \mathbb{P}\left(Y^1 = Y, Y^2 \neq Y\right)$, the probability that second sensor misclassifies an instance that is correctly classified by the first sensor. Strong dominance is the notion that suggests that this probability is zero. We find in these datasets that δ_{12} is small thus justifying our notion. In general we have found this behavior is representative of other cost-associated datasets. Note that strong dominance is not merely a consequence of improved accuracy with availability of more features. It is related to better *recall rates* of high-cost features relative to low-cost features.

We next show that strong dominance conditions ensures learnability. To this end, let $\Theta_{\text{SD}} = \{\theta \in \Theta_{\text{SA}} : \theta \text{ satisfies the strong dominance condition } \}$.

Theorem 1. *The set* Θ_{SD} *is learnable.*

We start with a proposition that will be useful beyond the proof of this result. In this proposition, $\gamma_i = \gamma_i(\theta)$ for $\theta \in \Theta_{\text{SA}}$ and $(Y, Y^1, \dots, Y^K) \sim \theta$.

Proposition 2. *For any* $i, j \in [K]$, $\gamma_i - \gamma_j = \mathbb{P}\left(Y^i \neq Y^j\right) - 2\mathbb{P}\left(Y^j \neq Y, Y^i = Y\right)$.

The proof motivates the definition of weak dominance, a concept that we develop next through a series of smaller propositions. In these propositions, as before $(Y, Y^1, \dots, Y^K) \sim P$ where $P \in M_1(\{0,1\}^{K+1})$, $\gamma_i = \mathbb{P}\left(Y^i \neq Y\right)$, $i \in [K]$, and $C_i = c_1 + \dots + c_i$. We start with a corollary of Proposition 2

Corollary 1. *Let* $i < j$. *Then* $0 \le \gamma_i - \gamma_j \le \mathbb{P}\left(Y^i \neq Y^j\right)$.

Proposition 3. *Let* $i < j$. *Assume*

$$C_j - C_i \notin [\gamma_i - \gamma_j, \mathbb{P}\left(Y^i \neq Y^j\right)). \tag{3}$$

Then $\gamma_i + C_i \le \gamma_j + C_j$ *if and only if* $C_j - C_i \ge \mathbb{P}\left(Y^i \neq Y^j\right)$.

Proof. $\Rightarrow$: From the premise, it follows that $C_j - C_i \ge \gamma_i - \gamma_j$. Thus, by (3), $C_j - C_i \ge \mathbb{P}\left(Y^i \neq Y^j\right)$. $\Leftarrow$: We have $C_j - C_i \ge \mathbb{P}\left(Y^i \neq Y^j\right) \ge \gamma_i - \gamma_j$, where the last inequality is by Corollary 1. □

Proposition 4. *Let* $j < i$. *Assume*

$$C_i - C_j \notin (\gamma_j - \gamma_i, \mathbb{P}\left(Y^i \neq Y^j\right)]. \tag{4}$$

Then, $\gamma_i + C_i \le \gamma_j + C_j$ *if and only if* $C_i - C_j \le \mathbb{P}\left(Y^i \neq Y^j\right)$.

Proof. $\Rightarrow$: The condition $\gamma_i + C_i \le \gamma_j + C_j$ implies that $\gamma_j - \gamma_i \ge C_i - C_j$. By Corollary 1 we get $\mathbb{P}\left(Y^i \neq Y^j\right) \ge C_i - C_j$. $\Leftarrow$: Let $C_i - C_j \le \mathbb{P}\left(Y^i \neq Y^j\right)$. Then, by (4), $C_i - C_j \le \gamma_j - \gamma_i$. □

These results motivate the following definition:

Definition 2 (Weak Dominance). *An instance* $\theta \in \Theta_{\text{SA}}$ *is said to satisfy the* weak dominance property *if for* $i = a^*(\theta)$,

$$\forall j > i \, : \, C_j - C_i \ge \mathbb{P}\left(Y^i \neq Y^j\right). \tag{5}$$

We denote the set of all instances in Θ_{SA} *that satisfies this condition by* Θ_{WD}.

Note that $\Theta_{\text{SD}} \subset \Theta_{\text{WD}}$ since for any $\theta \in \Theta_{\text{SD}}$, any $j > i = a^*(\theta)$, on the one hand $C_j - C_i \ge \gamma_i - \gamma_j$, while on the other hand, by the strong dominance property, $\mathbb{P}\left(Y^i \neq Y^j\right) = \gamma_i - \gamma_j$.

We now relate weak dominance to the optimality condition described in Eq. (1). Weak dominance can be viewed as a more stringent condition for optimal actions. Namely, for an action to be optimal we also require that the marginal cost be larger than marginal *absolute* error:

$$\underbrace{C_j - C_i}_{\text{Marginal Cost}} \ge \underbrace{E\left[\left|\mathbb{I}\{Y_t \neq Y_t^i\} - \mathbb{I}\{Y_t \neq Y_t^j\}\right|\right]}_{\text{Marginal Absolute Error}}, \ \forall j \ge i\,. \tag{6}$$

The difference between marginal error in Eq. (1) and marginal absolute error is the presence of the absolute value. We will show later that weak-dominant set is a maximal learnable set, namely, the set cannot be expanded while ensuring learnability.

We propose the following action selector $a_{\text{wd}} : M_1(\{0,1\}^K) \to [K]$:

Definition 3. *For* $P_S \in M_1(\{0,1\}^K)$ *let* $a_{\text{wd}}(P_S)$ *denote the smallest index* $i \in [K]$ *such that*

$$\forall j < i \, : \, C_i - C_j < \mathbb{P}\left(Y^i \neq Y^j\right), \tag{7a}$$

$$\forall j > i \, : \, C_j - C_i \ge \mathbb{P}\left(Y^i \neq Y^j\right), \tag{7b}$$

where $C_i = c_1 + \dots + c_i$, $i \in [K]$ *and* $(Y^1, \dots, Y^K) \sim P_S$. *(If no such index exists,* a_{wd} *is undefined, i.e.,* a_{wd} *is a partial function.)*

Proposition 5. *For any $\theta \in \Theta_{\text{WD}}$ with $\theta = P_S \otimes P_{Y|S}$, $a_{\text{wd}}(P_S)$ is well-defined.*

Proposition 6. *The map a_{wd} is sound over Θ_{WD}: In particular, for any $\theta \in \Theta_{\text{WD}}$ with $\theta = P_S \otimes P_{Y|S}$, $a_{\text{wd}}(P_S) = a^*(\theta)$.*

Corollary 2. *The set Θ_{WD} is learnable.*

Proof. By Proposition 5, a_{wd} is well-defined over Θ_{WD}, while by Proposition 6, a_{wd} is sound over Θ_{WD}. By Proposition 1, Θ_{WD} is learnable, as witnessed by a_{wd}. □

dd definitions
ts.. namely, a_{wd}
Θ_{WD}, a_{wd} sound

id this already!!

Proposition 7. *Let $\theta \in \Theta_{\text{SA}}$ and $\theta = P_S \otimes P_{Y|S}$ be such that a_{wd} is defined for P_S and $a_{\text{wd}}(P_S) = a^*(\theta)$. Then $\theta \in \Theta_{\text{WD}}$.*

Proof. Immediate from the definitions. □

An immediate corollary of the previous proposition is as follows:

Corollary 3. *Let $\theta \in \Theta_{\text{SA}}$ and $\theta = P_S \otimes P_{Y|S}$. Assume that a_{wd} is defined for P_S and $\theta \notin \Theta_{\text{WD}}$. Then $a_{\text{wd}}(P_S) \neq a^*(\theta)$.*

The next proposition states that a_{wd} is essentially the only sound action selector map defined for all instances derived from instances of Θ_{WD}:

Proposition 8. *Take any action selector map $a : M_1(\{0,1\}^K) \to [K]$ which is sound over Θ_{WD}. Then, for any P_S such that $\theta = P_S \otimes P_{Y|S} \in \Theta_{\text{WD}}$ with some $P_{Y|S}$, $a(P_S) = a_{\text{wd}}(P_S)$.*

The next result shows that the set Θ_{WD} is essentially a maximal learnable set in $\text{dom}(a_{\text{wd}})$:

Theorem 2. *Let $a : M_1(\{0,1\}^K) \to [K]$ be an action selector map such that a is sound over the instances of Θ_{WD}. Then there is no instance $\theta = P_S \otimes P_{Y|S} \in \Theta_{\text{SA}} \setminus \Theta_{\text{WD}}$ such that $P_S \in \text{dom}(a_{\text{wd}})$, the optimal action of θ is unique and $a(P_S) = a^*(\theta)$.*

nice to remove
assumption, but I
is could be made

Note that $\text{dom}(a_{\text{wd}}) \setminus \{P_S : \exists P_{Y|S} \text{ s.t. } P_S \otimes P_{Y|S} \in \Theta_{\text{WD}}\} \neq \emptyset$, i.e., the theorem statement is non-vacuous. In particular, for $K = 2$, consider (Y, Y^1, Y^2) such that Y and Y^1 are independent and $Y^2 = 1 - Y^1$, we can see that the resulting instance gives rise to P_S which is in the domain of a_{wd} for any $c \in \mathbb{R}_+^K$ (because here $\gamma_1 = \gamma_2 = 1/2$, thus $\gamma_1 - \gamma_2 = 0$ while $\mathbb{P}\left(Y^1 \neq Y^2\right) = 1$).

Proof. Let a as in the theorem statement. By Proposition 8, a_{wd} is the unique sound action-selector map over Θ_{WD}. Thus, for any $\theta = P_S \otimes P_{Y|S} \in \Theta_{\text{WD}}$, $a_{\text{wd}}(P_S) = a(P_S)$. Hence, the result follows from Corollary 3. □

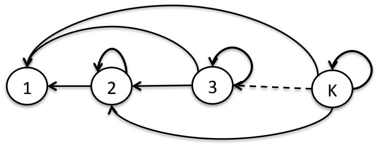

Figure 2: Neighborhood structure in bandit problem equivalent of SAP

While Θ_{WD} is learnable, it is not uniformly learnable, i.e., the minimax regret $\mathfrak{R}_n^*(\Theta_{\text{WD}}) = \inf_{\mathfrak{A}} \sup_{\theta \in \Theta_{\text{WD}}} \mathfrak{R}_n(\mathfrak{A}, \theta)$ over Θ_{WD} grows linearly:

Theorem 3. *Θ_{WD} is not uniformly learnable: $\mathfrak{R}_n^*(\Theta_{\text{WD}}) = \Omega(n)$.*

Proof.

6 Regret Equivalence

In this section we establish that SAP with strong dominance property is 'regret equivalent' to an instance of MAB with side-information and the corresponding algorithm for MAB can be suitably imported to solve SAP efficiently.

Let $\mathcal{P}_{\text{SAP}}$ be the set of SAPs with action set $\mathcal{A} = [K]$. The corresponding bandit problems will have the same action set, while for action $k \in [K]$ the neighborhood set is $\mathcal{N}(k) = [k]$. Take any instance $(P, c) \in \mathcal{P}_{\text{SAP}}$ and let $(Y, Y^1, \ldots, Y^K) \sim P$ be the unobserved state of environment in round s. We let the reward distribution for arm k in the corresponding bandit problem be a shifted Bernoulli distribution. In particular, the cost of arm k follows the distribution of $\mathbb{I}_{\{Y^k \neq Y^1\}} - C_k$ (we use costs here to avoid flipping signs).

M: We need
dit problem
not $\mathbb{I}_{\{Y^k \neq Y}$
mean of rev
We can ove
C_K to each
rewards are
won't chang
gret.

The costs for different arms are defined to be independent of each other. Let $\mathcal{P}_{\text{side}}$ denote the set of resulting bandit problems and let $f : \mathcal{P}_{\text{SAP}} \to \mathcal{P}_{\text{side}}$ be the map that transforms SAP instances to bandit instances by following the transformation that was just described.

Cs: Ok, so if they are independent of each other, then the joint distributions will not be same as if they were not independent of each other. Independence may lose information (e.g., may increase variance?). If we define them not to be independent of each other, we will need to be careful with the algorithms defined for bandits with side-observation: Do they use (in their proof) independence of rewards underlying different arms? I would think that they are not. The downside of not defining independent rewards is that the specification of bandits with side observations must allow this – complicating things a bit in the background. Another executive decision we should make is whether we like to see both costs and rewards.

M: We will
that review
asking this

Now let $\pi \in \Pi(\mathcal{P}_{\text{side}})$ be a policy for $\mathcal{P}_{\text{side}}$. Policy π can also be used on any (P, c) instance in $\mathcal{P}_{\text{SAP}}$ in an obvious way: In particular, given the history of actions and states $A_1, U_1, \ldots, A_t, U_t$ in $\theta = (P, c)$ where $U_s = (Y_s, Y_s^1, \ldots, Y_s^K)$ such that the distribution of U_s given that $A_s = a$ is P marginalized to $\mathcal{Y}^a$, the next action to be taken is $A_{t+1} \sim \pi(\cdot|A_1, V_1, \ldots, A_t, V_t)$,

e way rewards are
n SAP and ban-
he proof for regret
ase check.

where $V_s = (\mathbb{I}_{\{Y_s^1 \neq Y_s^1\}} - C_1, \ldots, \mathbb{I}_{\{Y_s^1 \neq Y_s^{A_s}\}} - C_{A_s})$. Let the resulting policy be denoted by π'. The following can be checked by simple direct calculation:

Proposition 9. *If $\theta \in \Theta_{\mathrm{SD}}$, then the regret of π on $f(\theta) \in \mathcal{P}_{\mathrm{side}}$ is the same as the regret of π' on θ.*

This implies that $\mathfrak{R}_T^*(\Theta_{\mathrm{SD}}) \leq \mathfrak{R}_T^*(f(\Theta_{\mathrm{SD}}))$.

Now note that this reasoning can also be repeated in the other "direction": For this, first note that the map f has a right inverse g (thus, $f \circ g$ is the identity over $\mathcal{P}_{\mathrm{side}}$) and if π' is a policy for $\mathcal{P}_{\mathrm{SAP}}$, then π' can be "used" on any instance $\theta \in \mathcal{P}_{\mathrm{side}}$ via the "inverse" of the above policy-transformation: Given the sequence $(A_1, V_1, \ldots, A_t, V_t)$ where $V_s = (B_s^1 + C_1, \ldots, B_s^K + C_s)$ is the vector of costs for round s with B_s^k being a Bernoulli with parameter γ_k, let $A_{t+1} \sim \pi'(\cdot|A_1, W_1, \ldots, A_t, W_t)$ where $W_s = (B_s^1, \ldots, B_s^{A_s})$. Let the resulting policy be denoted by π. Then the following holds:

Proposition 10. *Let $\theta \in f(\Theta_{\mathrm{SD}})$. Then the regret of policy π on $\theta \in f(\Theta_{\mathrm{SD}})$ is the same as the regret of policy π' on instance $f^{-1}(\theta)$.*

Hence, $\mathfrak{R}_T^*(f(\Theta_{\mathrm{SD}})) \leq \mathfrak{R}_T^*(\Theta_{\mathrm{SD}})$. In summary, we get the following result:

Corollary 4. $\mathfrak{R}_T^*(\Theta_{\mathrm{SD}}) = \mathfrak{R}_T^*(f(\Theta_{\mathrm{SD}}))$.

Cs: So this could in theory be used for upper and lower bounds.. However, $\mathcal{P}_{\mathrm{side}}$ is really special (because of the fixed costs) – hence it is unclear whether existing lower bounds, for example, would apply. The next step could be to describe policies for bandits with side observation starting from our paper with Yifan. We have two types of policies. One is asymptotically optimal, the other is minimax optimal. Can we have a single policy in our special problem that would be simultanously optimal in both cases? What happens when only weak dominance is satisfied?

7 Algorithms

The reduction of the previous section suggests that one can utilize an algorithm developed for stochastic bandits with side-observation to solve a SAP satisfying SD property. In this paper we make use of Algorithm 1 of [18]. While this algorithm was proposed for stochastic bandits with Gaussian side observations, as noted in the above paper, the algorithm is also suitable for problems where the payoff distributions are sub-Gaussian. As Bernoulli random variables are $\sigma^2 = 1/4$-sub-Gaussian (after centering), the algorithm is also applicable in our case.

g in the other pa-

Algorithm 1 Algorithm for SAP with SD property

```
Play action $K$ and observe $Y^1, \ldots, Y^K$.
Set $\hat{\gamma}_i^1 \leftarrow \mathbb{I}_{\{Y^1 \neq Y^i\}}$ for all $i \in [K]$.
Initialize the exploration count: $n_e \leftarrow 0$.
Initialize the allocation counts: $N_K(1) \leftarrow 1$.
for $t = 2, 3, \ldots$ do
    if $\frac{N(t-1)}{4\alpha \log t} \in C(\hat{\gamma}^{t-1})$ then
        Set $I_t \leftarrow \mathrm{argmin}_{k \in [K]} c(k, \hat{\gamma}^{t-1})$.
    else
        if $N_K(t-1) < \beta(n_e)/K$ then
            Set $I_t = K$.
        else
            Set $I_t$ to some $i$ for which
                $N_i(t-1) < u_i^*(\hat{\gamma}^{t-1}) 4\alpha \log t$.
        end if
        Increment exploration count: $n_e \leftarrow n_e + 1$.
    end if
    Play $I_t$ and observe $Y^1, \ldots, Y^{I_t}$.
    For $i \in [I_t]$, set
        $\hat{\gamma}_i^t \leftarrow (1 - 1/t)\hat{\gamma}_i^{t-1} + 1/t\, \mathbb{I}_{\{Y^1 \neq Y^i\}}$.
end for
```

For the convenience of the reader, we give the algorithm resulting from applying the reduction to Algorithm 1 of [18] in an explicit form. For specifying the algorithm we need some extra notation. Recall that given a SAP instance $\theta = (P, c)$, we let $\gamma_k = \mathbb{P}\left(Y \neq Y^k\right)$ where $(Y, Y^1, \ldots, Y^K) \sim P$ and $k \in [K]$. Let $k^* = \arg\min_k \gamma_k + C_k$ denote the optimal action and $\Delta_k(\theta) = \gamma_k + C_k - \gamma_{k^*} + C_{k^*}$ the sub-optimality gap of arm k. Further, let $\Delta^*(\theta) = \min\{\Delta_k(\theta), k \neq k^*\}$ denote the smallest positive sub-optimality gap and define $\Delta_k^*(\theta) = \max\{\Delta_k(\theta), \Delta^*(\theta)\}$.

Since cost vector c is fixed, in the following we use parameter γ in place of θ to denote the problem instance. A (fractional) allocation count $u \in [0, \infty)^K$ determines for each action i how many times the action is selected. Thanks to the cascade structure, using an action i implies observing the output of all the sensors with index j less than equal to i. Hence, a sensor j gets observed $u_j + u_{j+1} + \cdots + u_K$ times. We call an allocation count "sufficiently informative" if (with some level of confidence) it holds that *(i)* for each suboptimal choice, the number of observations for the corresponding sensor is sufficiently large to distinguish it from the optimal choice; and *(ii)* the optimal choice is also distinguishable from the second best choice. We collect these counts into the set $C(\gamma)$ for a given parameter γ: $C(\gamma) = \{u \in [0, \infty)^K : u_j + u_{j+1} + \cdots + u_K \geq \frac{2\sigma^2}{(\Delta_j^*(\theta))^2}, j \in [K]\}$ (note that $\sigma^2 = 1/4$). Further, let $u^*(\gamma)$ be the allocation count that minimizes the total expected excess cost over the set of sufficiently informative allocation counts:

In particular, we let $u^*(\gamma) = \text{argmin}_{u \in C(\gamma)} \langle u, \Delta(\theta) \rangle$ with the understanding that for any optimal action k, $u^*_k(\gamma) = \min\{u_k \,:\, u \in C(\gamma)\}$ (here, $\langle x, y \rangle = \sum_i x_i y_i$ is the standard inner product of vectors x, y). For an allocation count $u \in [0, \infty)^K$ let $m(u) \in \mathbb{N}^K$ denote total sensor observations, where $m_j(u) = \sum_{i=1}^{j} u_i$ corresponds to observations of sensor j.

The idea of the algorithm shown as Algorithm 1 is as follows: The algorithm keeps track of an estimate $\hat{\gamma}^t := (\hat{\gamma}^t_i)_{i \in [K]}$ of γ in each round, which is initialized by pulling arm K as this arm gives information about all the other arms. In each round, the algorithm first checks whether given the current estimate $\hat{\gamma}^t$ and the current confidence level (where the confidence level is gradually increased over time), the current allocation count $N(t) \in \mathbb{N}^K$ is sufficiently informative (cf. line 6). If this holds, the action that is optimal under $\hat{\gamma}(t)$ is chosen (cf. line 7). If the check fails, we need to explore. The idea of the exploration is that it tries to ensure that the "optimal plan" – assuming $\hat{\gamma}$ is the "correct" parameter – is followed (line 12). However, this is only reasonable, if all components of γ are relatively well-estimated. Thus, first the algorithm checks whether any of the components of γ has a chance of being extremely poorly estimated (line 9). Note that the requirement here is that a significant, but still altogether diminishing fraction of the *exploration rounds* is spent on estimating each components: In the long run, the fraction of exploration rounds amongst all rounds itself is diminishing; hence the forced exploration of line 10 overall has a small impact on the regret, while it allows to stabilize the algorithm.

For $\theta \in \Theta_{\text{SD}}$, let $\gamma(\theta)$ be the error probabilities for the various sensors. The following result follows from Theorem 6 of [18]:

Theorem 4. *Let $\epsilon > 0$, $\alpha > 2$ arbitrary and choose any non-decreasing $\beta(n)$ that satisfies $0 \leq \beta(n) \leq n/2$ and $\beta(m+n) \leq \beta(m) + \beta(n)$ for $m, n \in \mathbb{N}$. Then, for any $\theta \in \Theta_{\text{SD}}$, letting $\gamma = \gamma(\theta)$ the expected regret of Algorithm 1 after T steps satisfies*

eds to be checked..
$_{\max}(\theta)$ with 1.

$$R_T(\theta) \leq \Big(2K + 2 + \frac{4K}{\alpha - 2}\Big) + 4K \sum_{s=0}^{T} \exp\Big(\frac{-8\beta(s)\epsilon^2}{2K}\Big)$$

$$+ 2\beta\Big(4\alpha \log T \sum_{i \in [K]} u^*_i(\gamma, \epsilon) + K\Big) + 4\alpha \log T \sum_{i \in [K]} u^*_i(\gamma, \epsilon) d_i(\gamma),$$

where $u^*_i(\gamma, \epsilon) = \sup\{u^*_i(\gamma') \,:\, \|\gamma' - \gamma\|_\infty \leq \epsilon\}$.

Further specifying $\beta(n)$ and using the continuity of $u^*(\cdot)$ at θ, it immediately follows that Algorithm 1 achieves asymptotically optimal performance:

pasted this.
y have a lower

Corollary 5. *Suppose the conditions of Theorem 4 hold. Assume, furthermore, that $\beta(n)$ satisfies $\beta(n) = o(n)$ and $\sum_{s=0}^{\infty} \exp\Big(-\frac{\beta(s)\epsilon^2}{2K\sigma^2}\Big) < \infty$ for any $\epsilon > 0$, then for any θ such that $u^*(\theta)$ is unique,*

$$\limsup_{T \to \infty} R_T(\theta, c)/\log T \leq 4\alpha \inf_{u \in C_\theta} \langle u, d(\gamma(\theta)) \rangle \,.$$

Note that any $\beta(n) = an^b$ with $a \in (0, \frac{1}{2}]$, $b \in (0, 1)$ satisfies the requirements in Theorem 4 and Corollary 5.

The algorithm in 1 only estimates the disagreements $\mathcal{P}\{Y^1 \neq Y^j\}$ for all $j \in [K]$ which suffices to identify the optimal arm when the SAP satisfies the SD property (see Sec 6). Clearly, one can estimate pairwise disagreements probabilities $\mathcal{P}\{Y^i \neq Y^j\}$ for $i \neq j$ and use them to order the arms. We next develop an heuristic algorithm that uses this information and works for SAP satisfying WD property.

7.1 Algorithm for SAP with Weak Dominance

For any suboptimal arm $j < i^*$, $C_{i^*} - C_j \leq \mathcal{P}\{Y^{i^*} \neq Y^j\}$ holds, and when the WD property holds, we further have $C_j - C_{i^*} \geq \mathcal{P}\{Y^i \neq Y^j\}$ for all $j > i^*$. Thus, given the disagreement probabilities $\mathcal{P}\{Y^i \neq Y^j\}$ for all $i \neq j$, the set $\{i \in [K-1] \,:\, C_j - C_i \geq \mathcal{P}\{Y^i \neq Y^j\}$ for all $j > i\}$ includes the optimal arm. We use this idea in Algorithm 2 to identify the optimal arm when an instance of SAP satisfies the WD property. We will experimentally validate its performance on real datasets in the next sections.

Algorithm 2 Algorithm for SAP with WD property

```
1: Play action $K$ and observe $Y^1, \ldots, Y^K$
2: Set $\hat{\gamma}^1_{ij} \leftarrow \mathbb{I}_{\{Y^i \neq Y^j\}}$ for all $i, j \in [K]$ and $i < j$.
3: $n_i(1) \leftarrow \mathbb{I}_{\{i=K\}}$ $\forall i \in [K]$.
4: for $t = 2, 3, \ldots$ do
5:     $U^t_{ij} = \hat{\gamma}^{t-1}_{ij} + \sqrt{\frac{1.5\log(t)}{n_j(t-1)}}$ $\forall\, i, j \in [K]$ and $i < j$
6:     $S_t = \{i \in [K-1] : C_j - C_i \geq U^t_{ij}\ \forall\, j > i\}$
7:     Set $I_t = \arg\min S_t \cup \{K\}$
8:     Play $I_t$ and observe $Y^1, \ldots, Y^{I_t}$.
9:     for $i \in [I_t]$ do
10:        $n_i(t) \leftarrow n_i(t-1) + 1$
11:        $\hat{\gamma}^t_{ij} \leftarrow \Big(1 - \frac{1}{n_j(t)}\Big)\hat{\gamma}^{t-1}_{ij} + \frac{1}{n_j(t)}\,\mathbb{I}_{\{Y^j \neq Y^i\}} \forall\ i < j \leq I_t$
12:    end for
13: end for
```

The algorithm works as follows. It keeps track of $\hat{\gamma}^t_{ij}$ for all $i, j \in [K]$ and $i \neq j$ in each round, where $\hat{\gamma}_{ij}$ is an estimate of the probability $\mathcal{P}\{Y^i \neq Y^j\}$. In the first round, the algorithm plays arm K and initializes its values. In each subsequent round, the algorithm computes the upper confidence value of $\hat{\gamma}^t_{ij}$ denoted as U^t_{ij} (5) for all pairs (i, j) and orders the arms: i is considered better than arm j if $C_j - C_i \geq U^t_{ij}$. Specifically, the algorithm plays an arm i that satisfies $C_j - C_i \geq U^t_{ij}$ for all $j > i$ 6. If no such arm is found, then it plays arm K. $n_j(t), j \in [K]$ counts the total number of observation of pairs (Y^i, Y^j), for all $i < j$, till round t and uses it to update the estimates $\hat{\gamma}^t_{ij}$ (11).

8 Experiments

In this section we evaluate performance of 1 and 2 on synthetic and real datasets. For synthetic example, we consider data transmission over a binary symmetric channel, and for real world examples, we use diabetes (PIMA indiana) and heart disease (Clevland) from UCI dataset. In both datasets attributes/features are associated with costs, where features related to physical observations are cheap and that obtained from medical tests are costly. The experiments are setup as follows:

Synthetic: we consider data transmission over three binary symmetric channels (BSCs). Channel $i = 1, 2, 3$ flips input bit with probability p_i where $p_1 \geq p_2 \geq p_3$. Transmission over channel 1 is free and that over channel 2 and 3 costs price of c_2 and $c_3 \in (0, 1]$ units per bit, respectively. Input bits are generated with probability 0.7 and we set $p_1 = .3, p_2 = .1$ and $p_3 = .05$.

Datatsets: we obtain a sensor acquisition setup from the datasets as follows: Three svm classifiers (linear, $C = .01$) are trained for each dataset, first one using only cheap features, second one using cheap features plus few additional features and the third one using all features. These classifiers form sensors of a three stage SAP where classifier trained with cheap features is the first stage and that trained with all features forms the last stage. Cost of each stage is the sum of cost of features used to train that stage multiplied by a scaling factor λ (trade-off parameter for accuracy and costs). Specific details for each dataset is given below.

PIMA indians diabetes dataset consists of 768 instances and has 8 attributes. The labels identify if the instances are diabetic or not. 6 of the attributes (age, sex, triceps, etc.) obtained from physical observations are cheap, and 2 attributes (glucose and insulin) require expensive tests. First sensor of SAP is trained with 4 cheap attributes and costs \$4. Second sensor is trained from 6 attributes that cost \$6, and the last sesnor is trained with all 8 attributes that cost \$6. We set $C_1 = 4\lambda, C_2 = 6\lambda$ and $C_3 = 30\lambda$.

Heart disease dataset consists of 297 instance (without missing values) and has 13 attributes. 5 class labels $(0, 1, 2, 3, 4)$ are mapped to binary values by taking value 0 as 'absence' of disease and values $(1, 2, 3, 4)$ as 'presence' of disease. First senor of SAP is trained with 4 attributes which cost \$1 each and second sensor is trained with 8 attributes that cost \$1 each. Total cost of all attributes is \$568. We set $C_1 = 7\lambda, C_2 = 8\lambda$ and $C_3 = 568\lambda$.

dataset	γ_1	γ_2	γ_3	δ_{12}	δ_{13}	δ_{23}
BSC	.3	.1	.05	.07	.035	.045
diabetic	0.345	0.324	0.246	0.116	0.089	0.058
heart	0.292	0.27	0.146	0.124	0.067	0.079

Figure 7: Error statistics for real and synthetic datasets. $\delta_{ij} := \mathcal{P}\{Y^i = Y, Y_j \neq Y\}$.

Various error probabilities for synthetic and datasets are listed in Table (8). The probabilities for the datasets are computed on 40% hold out data. To run the online algorithm, an instance is randomly selected from the dataset (with replacement) in each round and is input to the algorithm. We repeat the experiments 20 times and average is shown in figures (8-8) with 95% confidence bounds. The left Figure in**??** we applied Algorithm 1 on BSC data that satisfies SD property. As see, the algorithm learns the optimal arm. In Figure 8 we applied Algorithm 2 on BSC data where we fixed $c_1 = 0$ and $c_2 + c_3 = 0.3$ while varying c_2 between 0.1 0.18. For all the cost values, arm 2 is optimal. When $c_2 \in \{0.1, 0.12, 0.14\}$ the weak dominance property is satisfied and the algorithm learns the optimal arm. For values $c_2 \in \{0.16, 0.18\}$ weak dominance property is violated we get linear regret as shown in the plots. In figure 8 we use the standard UCB algorithm that has access to the true labels and knows the if a sensor made error. As seen this setting learns much faster than than compared to unsupervised algorithm shown in figure 8 and 8. In figure we apply Algorithm 5 on diabetes dataset varying the trade-off factor λ. In the regions where WD property holds, the algorithm identifies the optimal arm.

9 Conclusions

We need to conclude soon.

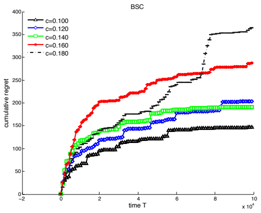

Figure 3: BSC wtih SD

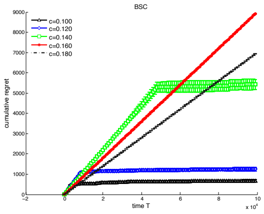

Figure 4: BSC dataset

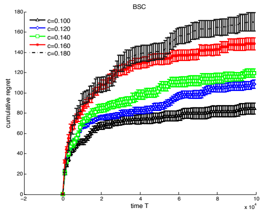

Figure 5: BSC dataset

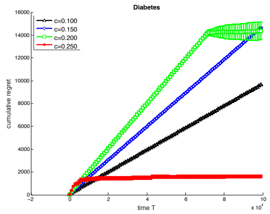

Figure 6: Heart dataset

In Fig. 8, BSC data is generated that satisfies SD property and Algorithm 1 is applied. In Fig. 8, Algorithm 2 is applied on BSC data. In Fig. 8, true labels are used and standard UCB algorithm is applied on BSC dataset. In Fig. 8, we applied Algorithm 5 on the heart dataset.

References

[1] K. Trapeznikov, V. Saligrama, and D. A. Castanon, "Multi-stage classifier design," *Machine Learning*, vol. 39, pp. 1–24, 2014.

[2] G. Bartók, D. Foster, D. Pál, A. Rakhlin, and C. Szepesvári, "Partial monitoring – classification, regret bounds, and algorithms," *Mathematics of Operations Research*, vol. 39, pp. 967–997, 2014.

[3] W. Thompson, "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples," *Biometrika*, vol. 25, pp. 285–294, 1933.

[4] K. Trapeznikov and V. Saligrama, "Supervised sequential classification under budget constraints," in *AISTATS*, 2013, pp. 235–242.

[5] J. Wang, K. Trapeznikov, and V. Saligrama, "Directed acyclic graph for resource constrained prediction," in *Proceeding of Conference on Neural Information Processing Systems, NIPS*, 2015.

[6] F. Nan, J. Wang, and V. Saligrama, "Feature-budgeted random forest," in *Proceeding of Conference on Neural Information Processing Systems, NIPS*, 2015.

[7] B. Póczos, Y. Abbasi-Yadkori, C. Szepesvári, R. Greiner, and N. Sturtevant, "Learning when to stop thinking and do something!" in *ICML*, 2009, pp. 825–832.

[8] R. Greiner, A. Grove, and D. Roth, "Learning cost-sensitive active classifiers," *Artificial Intelligence*, vol. 139, pp. 137–174, 2002.

[9] A. Kapoor and R. Greiner, "Learning and classifying under hard budgets," in *ECML*, 2005.

[10] B. Draper, J. Bins, and K. Baek, "Adore: Adaptive object recognition," in *International Conference on Vision Systems*, 1999, pp. 522–537.

[11] R. Isukapalli and R. Greiner, “Efficient interpretation policies,” in *International Joint Conference on Artificial Intelligence*, 2001, pp. 1381–1387.

[12] Y. Seldin, P. Bartlett, K. Crammer, and Y. Abbasi-Yadkori, “Prediction with limited advice and multiarmed bandits with paid observations,” in *Proceeding of International Conference on Machine Learning, ICML*, 2014, pp. 208–287.

[13] N. Zolghadr, G. Bartók, R. Greiner, A. György, and C. Szepesvári, “Online learning with costly features and labels,” in *NIPS*, 2013, pp. 1241–1249.

[14] R. Agrawal, D. Teneketzis, and V. Anantharam, “Asymptotically efficient adaptive allocation schemes for controlled i.i.d. processes: Finite parameter space,” *IEEE Transaction on Automatic Control*, vol. 34, pp. 258–267, 1989.

[15] S. Mannor and O. Shamir, “From bandits to experts: On the value of side-observations,” in *NIPS*, 2011.

[16] N. Alon, N. Cesa-Biancbi, O. Dekel, and T. Koren, “Online learning with feedback graphs:beyond bandits,” in *Proceeding of Conference on Learning Theory*, 2015, pp. 23–35.

[17] N. Alon, N. Cesa-Biancbi, C. Gentile, and Y. Mansour, “From bandits to experts: A tale of domination and independence,” in *Proceeding of Conference on Neural Information Processing Systems, NIPS*, 2013, pp. 1610–1618.

[18] Y. Wu, A. György, and C. Szepesvári, “Online learning with gaussian payoffs and side observations,” in *NIPS*, September 2015, pp. 1360–1368.

10 Appendix

Proof of Proposition 1

Proof. $\Rightarrow$: Let $\mathfrak{A}$ be an algorithm that achieves sublinear regret and pick an instance $\theta \in \Theta$. Let $P = P_S \otimes P_{Y|S}$. The regret $\mathfrak{R}_n(\mathfrak{A}, \theta)$ of $\mathfrak{A}$ on instance θ can be written in the form

$$\mathfrak{R}_n(\mathfrak{A}, \theta) = \sum_{k \in [K]} \mathbb{E}_{P_S}\left[N_k(n)\right] \Delta_k(\theta)\,,$$

where $N_k(n)$ is the number of times action k is chosen by $\mathfrak{A}$ during the n rounds while $\mathfrak{A}$ interacts with θ, $\Delta_k(\theta) = c(k, \theta) - c^*(\theta)$ is the immediate regret and $\mathbb{E}_{P_S}[\cdot]$ denotes the expectation under the distribution induced by P_S. In particular, $N_k(n)$ hides dependence on the iid sequence $Y_1, \ldots, Y_n \sim P_S$ that we are taking the expectation over here. Since the regret is sublinear, for any k suboptimal action, $\mathbb{E}_{P_S}[N_k(n)] = o(n)$. Define $a(P_S) = \min\{k \in [K]\,;\, \mathbb{E}_{P_S}[N_k(n)] = \Omega(n)\,\}$. Then, a is well-defined as the distribution of $N_k(n)$ for any k depends only on P_S (and c). Furthermore, $a(P_S)$ selects an optimal action.

$\Leftarrow$: Let a be the map in the statement and let $f : \mathbb{N}_+ \to \mathbb{N}_+$ be such that $1 \le f(n) \le n$ for any $n \in \mathbb{N}$, $f(n)/\log(n) \to \infty$ as $n \to \infty$ and $f(n)/n \to 0$ as $n \to \infty$ (say, $f(n) = \lceil \sqrt{n} \rceil$). Consider the algorithm that chooses $I_t = K$ for the first $f(n)$ steps, after which it estimates $\hat{P}_S$ by frequency counting and then uses $I_t = a(\hat{P}_S)$ in the remaining $n - f(n)$ trials. Pick any $\theta \in \Theta$ so that $\theta = P_S \otimes P_{Y|S}$. Note that by Hoeffding's inequality, $\sup_{y \in \{0,1\}^K} |\hat{P}_S(y) - P_S(y)| \le \sqrt{\frac{K \log(4n)}{2f(n)}}$ holds with probability $1 - 1/n$. Let n_0 be the first index such that for any $n \ge n_0$, $\sqrt{\frac{K \log(4n)}{2f(n)}} \le \Delta^*(\theta) \doteq \min_{k:\Delta_k(\theta)>0} \Delta_k(\theta)$. Such an index n_0 exists by our assumptions that f grows faster than $n \mapsto \log(n)$. For $n \ge n_0$, the expected regret of $\mathfrak{A}$ is at most $n \times 1/n + f(n)(1 - 1/n) \le 1 + f(n) = o(n)$. □

Proof of Proposition 8

Proof. Pick any $\theta = P_S \otimes P_{Y|S} \in \Theta_{\mathrm{WD}}$. If $A^*(\theta)$ is a singleton, then clearly $a(P_S) = a_{\mathrm{wd}}(P_S)$ since both are sound over Θ_{WD}. Hence, assume that $A^*(\theta)$ is not a singleton. Let $i = a^*(\theta) = \min A^*(\theta)$ and let $j = \min A^*(\theta) \setminus \{i\}$. We argue that $P_{Y|S}$ can be changed so that on the new instance i is still an optimal action, while j is not an optimal action, while the new instance $\theta' = P_S \otimes P'_{Y|S}$ is in Θ_{WD}.

The modification is as follows: Consider any $y^{-j} \doteq (y^1, \ldots, y^{j-1}, y^{j+1}, \ldots, y^K) \in \{0,1\}^{K-1}$. For $y, y^j \in \{0,1\}$, define $q(y|y^j) =$

$P_{Y|S}(y|y^1,\dots,y^{j-1},y^j,y^{j+1},\dots,y^K)$ and similarly let $q'(y|y^j) = P'_{Y|S}(y|y^1,\dots,y^{j-1},y^j,y^{j+1},\dots,y^K)$ Then, we let $q'(0|0)=0$ and $q'(0|1)=q(0|0)+q(0|1)$, while we let $q'(1|1)=0$ and $q'(1|0)=q(1|1)+q(1|0)$. This makes $P'_{Y|S}$ well-defined ($P'_{Y|S}(\cdot|y^1,\dots,y^K)$ is a distribution for any $y^1,\dots,y^K$). Further, we claim that the transformation has the property that it leaves γ_p unchanged for $p \neq j$, while γ_j is guaranteed to decrease. To see why γ_p is left unchanged for $p \neq j$ note that $\gamma_p = \sum_{y^p} P_{Y^p}(y^p)P_{Y|Y^p}(1-y^p|y^p)$. Clearly, P_{Y^p} is left unchanged. Introducing y^{-k} to denote a tuple where the kth component is left out, $P_{Y|Y^p}(1-y^p|y^p) = \sum_{y^{-p,-j}} P_{Y|Y^1,\dots,Y^K}(1-y^p|y^1,\dots,y^{j-1},0,y^{j+1},\dots,y^K) + P_{Y|Y^1,\dots,Y^K}(1-y^p|y^1,\dots,y^{j-1},1,y^{j+1},\dots,y^K)$ and by definition,

$$\begin{aligned}
&P_{Y|Y^1,\dots,Y^K}(1-y^p|y^1,\dots,y^{j-1},0,y^{j+1},\dots,y^K)\\
&\qquad + P_{Y|Y^1,\dots,Y^K}(1-y^p|y^1,\dots,y^{j-1},1,y^{j+1},\dots,y^K)\\
&\quad = P'_{Y|Y^1,\dots,Y^K}(1-y^p|y^1,\dots,y^{j-1},0,y^{j+1},\dots,y^K)\\
&\qquad + P'_{Y|Y^1,\dots,Y^K}(1-y^p|y^1,\dots,y^{j-1},1,y^{j+1},\dots,y^K)\,,
\end{aligned}$$

where the equality holds because "$q'(y|0)+q'(y|1) = q(y|0)+q(y|1)$". Thus, $P_{Y|Y^p}(1-y^p|y^p) = P'_{Y|Y^p}(1-y^p|y^p)$ as claimed. That γ_j is non-increasing follows with an analogue calculation. In fact, this shows that γ_j is strictly decreased if for any $(y^1,\dots,y^{j-1},y^{j+1},\dots,y^K) \in \{0,1\}^{K-1}$, either $q(0|0)$ or $q(1|1)$ was positive. If these are never positive, this means that $\gamma_j = 1$. But then j cannot be optimal since $c_j > 0$. Since j was optimal, γ_j is guaranteed to decrease.

Finally, it is clear that the new instance is still in Θ_{WD} since $a^*(\theta)$ is left unchanged. □

Proof of Proposition 5

Proof. Let $\theta \in \Theta_{\mathrm{WD}}$, $i = a^*(\theta)$. Obviously, (7b) holds by the definition of Θ_{WD}. Thus, the only question is whether (7a) also holds. We prove this by contradiction: Thus, assume that (7a) does not hold, i.e., for some $j < i$, $C_i - C_j \geq \mathbb{P}\left(Y^i \neq Y^j\right)$. Then, by Corollary 1, $\mathbb{P}\left(Y^i \neq Y^j\right) \geq \gamma_j - \gamma_i$, hence $\gamma_j + C_j \leq \gamma_i + C_i$, which contradicts the definition of i, thus finishing the proof. □

Proof of Proposition 6

Proof. Take any $\theta \in \Theta_{\mathrm{WD}}$ with $\theta = P_S \otimes P_{Y|S}$, $i = a_{\mathrm{wd}}(P_S)$, $j = a^*(\theta)$. If $i = j$, there is nothing to be proven. Hence, first assume that $j > i$. Then, by (7b), $C_j - C_i \geq \mathbb{P}\left(Y^i \neq Y^j\right)$. By Corollary 1, $\mathbb{P}\left(Y^i \neq Y^j\right) \geq \gamma_i - \gamma_j$. Combining these two inequalities we get that $\gamma_i + C_i \leq \gamma_j + C_j$, which contradicts with the definition of j. Now, assume that $j < i$. Then, by (5), $C_i - C_j \geq \mathbb{P}\left(Y^i \neq Y^j\right)$. However, by (7a), $C_i - C_j < \mathbb{P}\left(Y^i \neq Y^j\right)$, thus $j < i$ cannot hold either and we must have $i = j$. □

Proof of Proposition 2

Proof. We construct a map as required by Proposition 1. Take an instance $\theta \in \Theta_{\mathrm{WD}}$ and let $\theta = P_S \otimes P_{Y|S}$ be its decomposition as defined above. Let $\gamma_i = \mathbb{P}\left(Y^i \neq Y\right)$, $(Y, Y^1, \dots, Y^K) \sim \theta$. For identifying an optimal action in θ, it clearly suffices to know the sign of $\gamma_i + C_i - (\gamma_j + C_j)$ for all pairs $i, j \in [K]^2$. Since $C_i - C_j$ is known, it remains to study $\gamma_i - \gamma_j$. Without loss of generality (WLOG) let $i < j$. Then,

$$\begin{aligned}
0 \leq \gamma_i - \gamma_j &= \mathbb{P}\left(Y^i \neq Y\right) - \mathbb{P}\left(Y^j \neq Y\right)\\
&= \cancel{\mathbb{P}\left(Y^i \neq Y, Y^i = Y^j\right)} + \mathbb{P}\left(Y^i \neq Y, Y^i \neq Y^j\right) -\\
&- \left\{\cancel{\mathbb{P}\left(Y^j \neq Y, Y^i = Y^j\right)} + \mathbb{P}\left(Y^j \neq Y, Y^i \neq Y^j\right)\right\}\\
&= \mathbb{P}\left(Y^i \neq Y, Y^i \neq Y^j\right) + \mathbb{P}\left(Y^i = Y, Y^i \neq Y^j\right)\\
&- \left\{\mathbb{P}\left(Y^j \neq Y, Y^i \neq Y^j\right) + \mathbb{P}\left(Y^i = Y, Y^i \neq Y^j\right)\right\}\\
&\overset{(a)}{=} \mathbb{P}\left(Y^j \neq Y^i\right) - 2\mathbb{P}\left(Y^j \neq Y, Y^i = Y\right)\,,
\end{aligned}$$

where in (a) we used that $\mathbb{P}\left(Y^j \neq Y, Y^i \neq Y^j\right) = \mathbb{P}\left(Y^j \neq Y, Y^i = Y\right)$ and also $\mathbb{P}\left(Y^i = Y, Y^i \neq Y^j\right) = \mathbb{P}\left(Y^j \neq Y, Y^i = Y\right)$ which hold because Y, Y^i, Y^j only take on two possible values. □

Proof of Theorem 1

Proof of Theorem 1. We construct a map as required by Proposition 1. Take an instance $\theta \in \Theta_{\mathrm{SD}}$ and let $\theta = P_S \otimes P_{Y|S}$ be its decomposition as before. Let $\gamma_i = \mathbb{P}\left(Y^i \neq Y\right)$, $(Y, Y^1, \dots, Y^K) \sim \theta$, $C_i = c_1 + \dots + c_i$. For identifying an optimal action in θ, it clearly suffices to know the sign of $\gamma_i + C_i - (\gamma_j + C_j) = \gamma_i - \gamma_j + (C_i - C_j)$ for all pairs $i, j \in [K]^2$. Without loss of generality (WLOG) let $i < j$. By Proposition 2, $\gamma_i - \gamma_j = \mathbb{P}\left(Y^i \neq Y^j\right) - 2\mathbb{P}\left(Y^j \neq Y, Y^i = Y\right)$. Now, since θ satisfies the strong dominance condition, $\mathbb{P}\left(Y^j \neq Y, Y^i = Y\right) = 0$. Thus, $\gamma_i - \gamma_j = \mathbb{P}\left(Y^i \neq Y^j\right)$ which is a function of P_S only. Since $(C_i)_i$ are known, a map as required by Proposition 1 exists. □

Proof of Theorem 3

We first consider the case when $K = 2$ and arbitrarily choose $C_2 - C_1 = 1/4$. We will consider two instances, $\theta, \theta' \in \Theta_{\mathrm{WD}}$ such that for instance θ, action $k = 1$ is optimal with an action gap of $c(2,\theta) - c(1,\theta) = 1/4$ between the cost of the second and the first action, while for instance θ', $k = 2$ is the optimal action and

Cs: The th
be refined c

Instance θ		$Y^1 = Y^2$	$Y^1 \neq Y^2$
$Y^1 = Y$	$Y^2 = Y$	$\frac{3}{8}$	0
	$Y^2 \neq Y$	0	$\frac{1}{8}$
$Y^1 \neq Y$	$Y^2 = Y$	0	$\frac{1}{8}$
	$Y^2 \neq Y$	$\frac{3}{8}$	0

Instance θ'		$Y^1 = Y^2$	$Y^1 \neq Y^2$
$Y^1 = Y$	$Y^2 = Y$	$\frac{3}{8} - \epsilon$	0
	$Y^2 \neq Y$	0	0
$Y^1 \neq Y$	$Y^2 = Y$	0	$\frac{2}{8} + \epsilon$
	$Y^2 \neq Y$	$\frac{3}{8}$	0

Table 2: The construction of two problem instances for the proof of Theorem 3.

	θ	θ'
$\gamma_1 = \mathbb{P}\left(Y^1 \neq Y\right)$	$\frac{1}{4}$	$\frac{5}{8} + \epsilon$
$\gamma_2 = \mathbb{P}\left(Y^2 \neq Y\right)$	$\frac{1}{4}$	$\frac{3}{8}$
$\gamma_2 \leq \gamma_1{}^{(*)}$	✓	✓
$c(1, \cdot)$	$\frac{1}{4}$	$\frac{5}{8} + \epsilon$
$c(2, \cdot)$	$\frac{2}{4}$	$\frac{5}{8}$
$a^*(\cdot)$	$k = 1$	$k = 2$
$\mathbb{P}\left(Y^1 \neq Y^2\right)$	$\frac{1}{4}$	$\frac{1}{4} + \epsilon$
$\theta \in \Theta_{\mathrm{WD}}{}^{(**)}$	$\frac{1}{4} \geq \frac{1}{4}$ ✓	✓
$\lvert c(1, \cdot) - (2, \cdot) \rvert$	$\frac{1}{4}$	ϵ

Table 3: Calculations for the proof of Theorem 3.

the action gap is $c(1, \theta) - c(2, \theta) = \epsilon$ where $0 < \epsilon < 3/8$. Further, the entries in $P_S(\theta)$ and $P_S(\theta')$ differ by at most ϵ. From this, a standard reasoning gives that no algorithm can achieve sublinear minimax regret over Θ_{WD} because any algorithm is only able to identify P_S.

n of $P_S(\theta)$ early
good idea to add
tion too for the
Θ to P_S. Also, we
remove c from
nition; in every
oning for a fixed c,
fluous to keep c in
nition.

ning, it is now
c is fixed and
noved from defini-
instance now. Will
ojection notation
d of edits.

The constructions of θ and θ' are shown in Table 2: The entry in a cell gives the probability of the event as specified by the column and row labels. For example, in instance θ, 3/8 is the probability of $Y = Y^1 = Y^2$, while the probability of $Y^1 = Y \neq Y^2$ is 1/8. Note that the cells with zero actually correspond to impossible events, i.e., these cannot be assigned a positive probability. The rationale of a redundant (and hence sparse) table is so that probabilities of certain events of interest, such as $Y^1 \neq Y^2$ are easier to determine based on the table. The reader should also verify that the positive probabilities correspond to events that are possible.

We need to verify the following: *(i)* $\theta, \theta' \in \Theta_{\mathrm{WD}}$; *(ii)* the optimality of the respective actions in the respective instances; *(iii)* the claim concerning the size of the action gaps; *(iv)* that $P_S(\theta)$ and $P_S(\theta')$ are close. Details of the calculations to support *(i)*–*(iii)* can be found in Table 3. The row marked by $(*)$ supports that the instances are proper SAP instances. In the row marked by $(**)$, there is no requirement for θ' because in θ' action two is optimal, and hence there is no action with larger index than the optimal action, hence $\theta' \in \Theta_{\mathrm{WD}}$ automatically holds. To verify the closeness of $P_S(\theta)$ and $P_S(\theta')$ we actually would need to first specify P_S (the tables do not fully specify these). However, it is clear the only restriction we put on P_S is the value of $\mathbb{P}\left(Y^1 \neq Y^2\right)$ (and that of $\mathbb{P}\left(Y^1 = Y^2\right)$) and these values are within an ϵ distance of each other. Hence, P_S can also be specified to satisfy this. In particular, one possibility for P and P_S are given in Table 4.

□

Proof of Proposition 9

Proof. First note that the mapping of the policies is such that number of pull of arm k after n rounds by policy π on problem instance $f(\theta)$ is the same as the number of pulls of arm k by π' on problem instance θ. Recall that mean value of arm k in problem instance θ $is \gamma_k + C_k$ and that of corresponding arm in problem instance $f(\theta)$ is $\gamma_1 - (\gamma_i + C_i)$. We have

$$\mathfrak{R}_n(\pi', \theta) = \sum_{k \in [K]} \mathbb{E}_{P_S}\left[N_k(n)\right]\left(\gamma_k + C_k - \gamma_{k^*} - C_{k^*}\right),$$

and

$$\begin{aligned}
&\mathfrak{R}_n(\pi, f(\theta)) \\
&= \sum_{k \in [K]} \mathbb{E}_{P_S}\left[N_k(n)\right]\left(\max_{i \in [K]}\{\gamma_1 - \gamma_i - C_i\} - (\gamma_1 - \gamma_k - C_k)\right) \\
&= \sum_{k \in [K]} \mathbb{E}_{P_S}\left[N_k(n)\right]\left(\gamma_k + C_k - \min_{i \in [K]}\{\gamma_i + C_i\}\right) \\
&= \mathfrak{R}_n(\pi', \theta).
\end{aligned}$$

□

Y^1	Y^2	Y	θ	θ'
0	0	0	$\frac{3}{8}$	$\frac{3}{8} - \epsilon$
0	0	1	$\frac{3}{8}$	$\frac{3}{8} - \epsilon$
0	1	0	0	0
0	1	1	0	0
1	0	0	$\frac{1}{8}$	$\frac{2}{8} + \epsilon$
1	0	1	$\frac{1}{8}$	0
1	1	0	0	0
1	1	1	0	0

Y^1	Y^2	θ	θ'
0	0	$\frac{6}{8}$	$\frac{6}{8} - \epsilon$
0	1	0	0
1	0	$\frac{2}{8}$	$\frac{2}{8} + \epsilon$
1	1	0	0

Table 4: Probability distributions for instances θ and θ'. On the left are shown the joint probability distributions, while on the right are shown their marginals for the sensors.