The Complexity of Tracking a Stopping Time

Urs Niesen Aslan Tchamkerten Gregory Wornell
Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science
Cambridge, MA
Email: {uniesen,tcham,gww}@mit.edu

***Abstract*— We present a generalization of the well-known Bayesian change-point detection problem. Specifically, let $\{(X_i, Y_i)\}_{i\geq 1}$ be a sequence of pairs of random variables, and let S be a stopping time with respect to $\{X_i\}_{i\geq 1}$. We assume that the (X_i, Y_i)'s take values in the same finite alphabet $\mathcal{X}\times\mathcal{Y}$. For a fixed $\kappa \geq 1$, we consider the problem of finding a stopping time $T \leq \kappa$ with respect to $\{Y_i\}_{i\geq 1}$ that optimally tracks S, in the sense that T minimizes the average *reaction time* $\mathbb{E}(T-S)^+$, while it keeps the *false-alarm probability* $\mathbb{P}(T < S)$ below a given threshold $\alpha \in [0, 1]$.**

In previous work, we presented an algorithm that computes the optimal expected reaction times for all $\alpha \in [0, 1]$ such that $\alpha \geq \mathbb{P}(S > \kappa)$, and constructs the associated optimal stopping times T. In this paper, we provide a sufficient condition on $\{(X_i, Y_i)\}_{i\geq 1}$ and S under which the algorithm running time is polynomial in κ, and we illustrate this condition with two examples: a Bayesian change-point problem and a pure tracking stopping time problem.

I. Problem statement and examples

The tracking stopping time (TST) problem is defined as follows. Let $\{(X_i, Y_i)\}_{i\geq 1}$ be a sequence of pairs of random variables. Alice observes $X_1, X_2, \ldots$ and chooses a stopping time[1] (s.t.) S with respect to that sequence. Knowing the distribution of $\{(X_i, Y_i)\}_{i\geq 1}$ and the stopping rule S, but having access only to the Y_i's, Bob wishes to find a time T that stops as closely as possible to S. Specifically, Bob aims to find a s.t. T minimizing the expected delay $\mathbb{E}(T-S)^+ \triangleq \mathbb{E}\max\{0, T-S\}$, while keeping the false-alarm probability $\mathbb{P}(T < S)$ below a certain threshold $\alpha \in [0, 1]$.

This problem finds applications, for instance, in monitoring, forecasting, and communication [1].

In the following subsection we discuss the relationship between the TST problem and the Bayesian change-point problem.

The Bayesian change-point detection as a TST problem

The Bayesian change-point problem can be formulated as follows. Let θ be a random variable taking values over the positive integers. Let $\{Y_i\}_{i\geq 1}$ be a sequence of random variables such that, given the value of θ, the conditional probability of Y_n given Y^{n-1} is $P_0(\cdot|Y^{n-1})$ for $n < \theta$ and is $P_1(\cdot|Y^{n-1})$ for $n \geq \theta$. The problem is to find a s.t. T with respect to the Y_i's minimizing the change-point reaction delay $\mathbb{E}(T-\theta)^+$ while keeping the false-alarm probability $\mathbb{P}(T < \theta)$ below a certain threshold $\alpha \in [0, 1]$.

Shiryaev [2, Chapter 4.3] considered the Lagrangian formulation of the above problem and aimed to minimize

$$J_\lambda(T) \triangleq \mathbb{E}(T-S)^+ + \lambda\mathbb{P}(T < S)$$

among all s.t.'s T, for fixed $\lambda \geq 0$. Assuming a geometric prior on the change-point θ and that before and after θ the observations are independent with common density function f_0 for $t < \theta$ and f_1 for $t \geq \theta$, Shiryaev showed that the optimal T stops as soon as the posterior probability that a change occurred exceeds a certain fixed threshold. Later Yakir [3] generalized Shiryaev's result by considering finite-state Markov chains. For more general prior distributions on θ the problem is known to become difficult to handle. However, in the limit of small false-alarm probabilities, Lai [4] derived asymptotic optimal detection policies for the Bayesian change-point problem under general assumptions on the distributions on the change-point and on the observed process.

As we shall see, the TST problem represents a generalization of the Bayesian change-point problem. Interestingly, even though easy computable solutions for the Bayesian change-point problem have been found only for specific cases we shall present certain non trivial TST instances that also admit easy computable solutions. We use here the terminology 'computable' since our approach is indeed algorithmic.

To see that the Bayesian change-point problem can be formulated as a TST problem, it suffices to define the sequence of binary random variables $\{X_i\}_{i\geq 1}$ such that $X_i = 0$ if $i < \theta$ and $X_i = 1$ if $i \geq \theta$, and to define the stopping time $S \triangleq \inf\{i : X_i = 1\}$ (i.e., $S = \theta$). The change-point problem now becomes a TST problem where the goal is to track S having access only to the Y_i's.

In general, however, the TST problem cannot be formulated as a Bayesian change-point problem. Indeed, for the Bayesian change-point problem we have for any $k > n$

$$\begin{aligned}\mathbb{P}(\theta = k|Y^n = y^n, \theta > n) &= \frac{\mathbb{P}(Y^n = y^n, \theta > n|\theta = k)\mathbb{P}(\theta = k)}{\mathbb{P}(Y^n = y^n|\theta > n)\mathbb{P}(\theta > n)} \\ &= \frac{\mathbb{P}(Y^n = y^n|\theta = k)\mathbb{P}(\theta = k)}{\mathbb{P}(Y^n = y^n|\theta > n)\mathbb{P}(\theta > n)}\end{aligned}$$

This work was supported in part by NSF under Grant No. CCF-0515122, and by a University IR&D Grant from Draper Laboratory.

[1] An integer-valued random variable S is called a s.t. with respect to a sequence of random variables $\{X_i\}_{i\geq 1}$ if, conditioned on $\{X_i\}_{i=1}^n$, the event $\{S = n\}$ is independent of $\{X_i\}_{i=n+1}^\infty$, for all $n \geq 1$.

$$= \mathbb{P}(\theta = k | \theta > n) \quad (1)$$

since $\mathbb{P}(Y^n = y^n | \theta = k) = \mathbb{P}(Y^n = y^n | \theta > n)$. Hence, conditioned on the event $\{\theta > n\}$, the first n observations Y^n are independent of θ. In other words, given that no change occurred up to time n, the observations y^n are useless in predicting the value of the change point θ. In contrast, for the TST problem, in general we have

$$\mathbb{P}(S = k | Y^n = y^n, S > n) \neq \mathbb{P}(S = k | S > n) \quad (2)$$

since $\mathbb{P}(Y^n = y^n | S = k)$ may not be equal to $\mathbb{P}(Y^n = y^n | S > n)$.

This paper is organized as follows. In Section II we formally define the TST optimization problem, and in Section III we recall the algorithm solving it [5]. In Section IV we provide conditions under which the algorithm has low complexity and illustrate this in Section V with two examples.

II. The optimization problem

Let $\{(X_i, Y_i)\}_{i \geq 1}$ be a discrete-time process where the X_i's and Y_i's take value in some finite alphabets $\mathcal{X}$ and $\mathcal{Y}$, respectively. Let S be a s.t. with respect to $\{X_i\}_{i \geq 1}$ such that $\mathbb{P}(S < \infty) = 1$, and let $\kappa \geq 1$. For any α such that $\mathbb{P}(S > \kappa) \leq \alpha \leq 1$ we aim to find

$$d(\alpha) \triangleq \min_{\substack{T : \mathbb{P}(T < S) \leq \alpha \\ T \leq \kappa}} \mathbb{E}(T - S)^+ \quad (3)$$

where the minimization is over all (possibly randomized[2]) s.t.'s with respect to $\{Y_i\}_{i \geq 1}$. The mild constraint $T \leq \kappa$ is motivated at the end of this section. Now, the extreme points of the set of all s.t.'s over $\{Y_i\}_{i \geq 1}$ are non-randomized s.t.'s [6], [7]. This means that any randomized s.t. $T \leq \kappa$ can be written as a finite convex combination of non-randomized s.t.'s $\{T_m\}$, i.e.

$$\mathbb{P}(T = k) = \sum_m \mathbb{P}(T_m = k) a_m$$

for any integer k, where $a_m \geq 0$ and $\sum_m a_m = 1$. This implies that

$$\mathbb{P}(T < S) = \sum_m a_m \mathbb{P}(T_m < S),$$
$$\mathbb{E}(T - S)^+ = \sum_m a_m \mathbb{E}(T_m - S)^+,$$

i.e., $\mathbb{P}(T < S)$ and $\mathbb{E}(T - S)^+$ are linear with respect to T. Therefore the epigraph of $d(\alpha)$ is convex, and its extreme points are achieved by non-randomized s.t.'s. Since there are only a finite number of non-randomized s.t.'s bounded by κ, the function $d(\alpha)$ is piecewise linear. The typical shape of $d(\alpha)$ is depicted in Figure 1, where the break-points are achieved by non-randomized s.t.'s.

Using Lagrange duality yields

$$d(\alpha) = \sup_{\lambda \geq 0} \min_{T \leq \kappa} \left(J_\lambda(T) - \lambda \alpha \right), \quad (4)$$

[2]A s.t. T is *non-randomized* if $\mathbb{P}(T = n | Y^n = y^n) \in \{0, 1\}$ for all $y^n \in \mathcal{Y}^n$ and $n \geq 1$. In contrast a s.t. T is *randomized* if $\mathbb{P}(T = n | Y^n = y^n) \in [0, 1]$ for all $y^n \in \mathcal{Y}^n$ and $n \geq 1$.

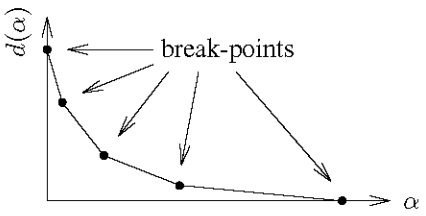

Fig. 1. Typical shape of the expected delay $d(\alpha)$ as a function of false alarm probability α.

where

$$J_\lambda(T) \triangleq \mathbb{E}(T - S)^+ + \lambda \mathbb{P}(T < S) \, .$$

Without loss of optimality, we may restrict the minimization in (4) to be over the set of s.t.'s that represent the extreme points of the epigraph of $d(\alpha)$, i.e., the non-randomized s.t.'s bounded by κ.

The restriction $T \leq \kappa$ in the minimization (3) is motivated as follows. Let T^* be the s.t. that minimizes $\mathbb{E}(T - S)^+$ subject to $\mathbb{P}(T < S) \leq \alpha$, but not necessarily bounded by κ. Assume first $S \leq \kappa$ for some integer $\kappa \geq 1$. In this case the restriction $T \leq \kappa$ in (3) is without loss of optimality since $T^* \leq \kappa$. Second, assume that S is unbounded but still satisfies $\mathbb{P}(S < \infty) = 1$. Let us choose κ such that $\mathbb{P}(S \geq \kappa) \leq \delta$, and set $\widetilde{T} \triangleq \min\{T^*, \kappa\}$. Note first that $\mathbb{E}(\widetilde{T} - S)^+ \leq \mathbb{E}(T^* - S)^+$. For the false-alarm probability, we have

$$\begin{aligned} \mathbb{P}(\widetilde{T} < S) &= \mathbb{P}(\widetilde{T} < S, T^* \geq \kappa) + \mathbb{P}(\widetilde{T} < S, T^* < \kappa) \\ &\leq \mathbb{P}(S > \kappa) + \mathbb{P}(T^* < S) \\ &\leq \delta + \alpha \, . \end{aligned}$$

Therefore the bounded s.t. $\widetilde{T}$ yields an approximation to T^* in the sense that it gives an expected reaction delay at least as good as T^*, while having only a slightly higher false-alarm probability. From now on, and unless stated otherwise, we assume S to be bounded by some $\kappa \geq 1$.

III. An algorithm for computing $d(\alpha)$

We first establish a few preliminary results later used to evaluate $\min_{T \leq \kappa} J_\lambda(T)$. Emphasis is put on the finite tree representation of bounded s.t.'s with respect to finite alphabet processes.

Let us introduce a few notational conventions. The set $\mathcal{Y}^*$ represents the set of all finite sequences over $\mathcal{Y}$. An element in $\mathcal{Y}^*$ is denoted either by $\boldsymbol{y}$ or by y^n, depending on whether we want to emphasize the length of the sequence or not. To any non-randomized s.t. T we associate a unique $|\mathcal{Y}|$-ary tree $\mathcal{T}$ — i.e., all the nodes of $\mathcal{T}$ have either zero or exactly $|\mathcal{Y}|$ children — having each node specified by some $\boldsymbol{y} \in \mathcal{Y}^*$, where $\rho \boldsymbol{y}$ represents the vertex path from the root ρ to the node $\boldsymbol{y}$. The depth of a node $y^n \in \mathcal{T}$ is denoted by $l(y^n) \triangleq n$. The tree consisting only of the root is the trivial tree. A node $y^n \in \mathcal{T}$ is a leaf if $\mathbb{P}(T = n | Y^n = y^n) = 1$. We denote by $\mathcal{L}(\mathcal{T})$ the leaves of $\mathcal{T}$ and by $\mathcal{I}(\mathcal{T})$ the intermediate (or non-terminal) nodes of $\mathcal{T}$. The notation $T(\mathcal{T})$ is used to denote

the s.t. T induced by the tree $\mathcal{T}$. Given a node $\boldsymbol{y}$ in $\mathcal{T}$, let $\mathcal{T}_y$ be the subtree of $\mathcal{T}$ rooted in $\boldsymbol{y}$. Finally let $\mathcal{D}(\mathcal{T}_y)$ denote the descendants of $\boldsymbol{y}$ in $\mathcal{T}$. The next example illustrates these notations.

Example 1. Let $\mathcal{Y} = \{0, 1\}$ and $\kappa = 2$. The tree $\mathcal{T}$ depicted in Figure 2 corresponds to the non-randomized s.t. T taking value one if $Y_1 = 1$ and value 2 if $Y_1 = 0$. The sets $\mathcal{L}(\mathcal{T})$ and $\mathcal{I}(\mathcal{T})$ are given by $\{00, 01, 1\}$ and $\{\rho, 0\}$, respectively. The subtree $\mathcal{T}_0$ of $\mathcal{T}$ consists of the nodes $\{0, 00, 01\}$, and its descendants $\mathcal{D}(\mathcal{T}_0)$ are $\{00, 01\}$. The subtree $\mathcal{T}_\rho$ is the same as $\mathcal{T}$, and its descendants $\mathcal{D}(\mathcal{T}_\rho)$ are $\{0, 1, 00, 01\}$. $\Diamond$

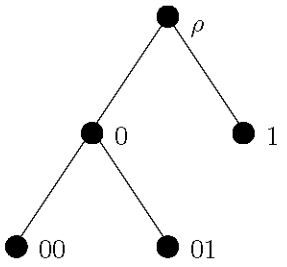

Fig. 2. Tree corresponding to the s.t. T defined by $T = 1$ if $Y_1 = 1$, and $T = 2$ else.

Below we describe an algorithm that, for a given s.t. S, constructs a sequence of s.t.'s $\{T(\mathcal{T}^m)\}_{m=0}^M$ and Lagrange multipliers $\{\lambda_m\}_{m=0}^M$ with the following two properties. First, the $\mathcal{T}^m$'s and λ_m's are ordered in the sense that $\mathcal{T}^M \subset \mathcal{T}^{M-1} \subset \ldots \subset \mathcal{T}^0$ and $0 = \lambda_M \leq \lambda_{M-1} \leq \ldots \leq \lambda_1 \leq \lambda_0 = \infty$. (Here the symbol $\subset$ denotes inclusion, not necessarily strict.) Second, for any $m \in \{0, \ldots, M\}$ and $\lambda \in (\lambda_m, \lambda_{m-1}]$ the tree $\mathcal{T}^{m-1}$ minimizes $J_\lambda(T) \triangleq J_\lambda(T(\mathcal{T}))$ among all non-randomized s.t.'s. The algorithm builds upon ideas from the CART algorithm for the construction of classification and regression trees [8, Chapter 10].

Before we state the algorithm, we need to introduce a few quantities. Given a s.t. T represented by its $|\mathcal{Y}|$-ary tree $\mathcal{T}$, we have

$$\begin{aligned} J_\lambda(T) &= \mathbb{E}(T - S)^+ + \lambda \mathbb{P}(T < S) \\ &= \sum_{\boldsymbol{y} \in \mathcal{L}(\mathcal{T})} \mathbb{P}(\boldsymbol{Y} = \boldsymbol{y}) \\ &\quad \times \Big(\mathbb{E}\big((l(\boldsymbol{y}) - S)^+ | \boldsymbol{Y} = \boldsymbol{y}\big) + \lambda \mathbb{P}\big(S > l(\boldsymbol{y}) | \boldsymbol{Y} = \boldsymbol{y}\big) \Big) \\ &= \sum_{\boldsymbol{y} \in \mathcal{L}(\mathcal{T})} b(\boldsymbol{y}) + \lambda a(\boldsymbol{y}) \\ &= \sum_{\boldsymbol{y} \in \mathcal{L}(\mathcal{T})} J_\lambda(\boldsymbol{y}), \end{aligned}$$

where

$$\begin{aligned} a(\boldsymbol{y}) &\triangleq \mathbb{P}(\boldsymbol{Y} = \boldsymbol{y}) \mathbb{P}(S > l(\boldsymbol{y}) | \boldsymbol{Y} = \boldsymbol{y}), \\ b(\boldsymbol{y}) &\triangleq \mathbb{P}(\boldsymbol{Y} = \boldsymbol{y}) \mathbb{E}\big((l(\boldsymbol{y}) - S)^+ | \boldsymbol{Y} = \boldsymbol{y}\big), \\ J_\lambda(\boldsymbol{y}) &\triangleq b(\boldsymbol{y}) + \lambda a(\boldsymbol{y}) \ . \end{aligned}$$

We extend the definition of $J_\lambda(\cdot)$ to subtrees of $\mathcal{T}$ by setting $J_\lambda(\mathcal{T}_y) \triangleq \sum_{\gamma \in \mathcal{L}(\mathcal{T}_y)} J_\lambda(\gamma)$. With this definition[3]

$$J_\lambda(\mathcal{T}_y) = \begin{cases} J_\lambda(\boldsymbol{y}) & \text{if } \boldsymbol{y} \in \mathcal{L}(\mathcal{T}), \\ \sum_{\gamma \in \mathcal{Y}} J_\lambda(\mathcal{T}_{y\gamma}) & \text{if } \boldsymbol{y} \in \mathcal{I}(\mathcal{T}). \end{cases}$$

Similarly, we define $a(\mathcal{T}_y) \triangleq \sum_{\gamma \in \mathcal{L}(\mathcal{T}_y)} a(\gamma)$ and $b(\mathcal{T}_y) \triangleq \sum_{\gamma \in \mathcal{L}(\mathcal{T}_y)} b(\gamma)$.

For a given $\lambda \geq 0$ and $\mathcal{T}$, define $\mathcal{T}(\lambda) \subset \mathcal{T}$ to be the subtree of $\mathcal{T}$ such that $J_\lambda(\mathcal{T}(\lambda)) \leq J_\lambda(\mathcal{T}')$ for all subtrees $\mathcal{T}' \subset \mathcal{T}$, and such that $\mathcal{T}(\lambda) \subset \mathcal{T}'$ for all subtrees $\mathcal{T}' \subset \mathcal{T}$ satisfying $J_\lambda(\mathcal{T}(\lambda)) = J_\lambda(\mathcal{T}')$. In words, among all subtrees of $\mathcal{T}$ yielding a minimal cost for a given λ, the tree $\mathcal{T}(\lambda)$ is the smallest. It can be shown that such a smallest optimal subtree always exists, and hence $\mathcal{T}(\lambda)$ is well defined.

Define for any $\boldsymbol{y} \in \mathcal{I}(\mathcal{T})$

$$g(\boldsymbol{y}, \mathcal{T}) \triangleq \frac{b(\mathcal{T}_y) - b(\boldsymbol{y})}{a(\boldsymbol{y}) - a(\mathcal{T}_y)} \ .$$

The following algorithm fully characterizes $d(\alpha)$ by computing its set of break-points [5].

Algorithm Compute the break-points $\{\alpha_m, d_m\}_{m=0}^M$ of $d(\alpha)$.

$\mathcal{T}^0 \Leftarrow$ complete tree of depth κ
$\lambda_0 \Leftarrow \infty$
$m \Leftarrow 0$
repeat
  $m \Leftarrow m + 1$
  $\lambda_m \Leftarrow \max_{\boldsymbol{y} \in \mathcal{I}(\mathcal{T}^{m-1})} g\big(\boldsymbol{y}, \mathcal{T}^{m-1}\big)$
  $\mathcal{T}^m \Leftarrow \mathcal{T}^{m-1} \setminus \bigcup_{\substack{\boldsymbol{y} \in \mathcal{I}(\mathcal{T}^{m-1}): \\ g(y, \mathcal{T}^{m-1}) = \lambda_m}} \mathcal{D}(\mathcal{T}_y^{m-1})$
  $\alpha_m \Leftarrow \mathbb{P}(T(\mathcal{T}^m) < S)$
  $d_m \Leftarrow \mathbb{E}(T(\mathcal{T}^m) - S)^+$
until $\mathcal{T}^m = \{\rho\}$

As a $|\mathcal{Y}|$-ary tree has less than $|\mathcal{Y}|^\kappa$ non-terminal nodes, the algorithm terminates after at most that many iterations. Further, one may check that each iteration has a running time that is $\exp(O(\kappa))$. Therefore, the worst case running time of the algorithm is $\exp(O(\kappa))$. This is to be compared, for instance, with exhaustive search that has a $\Omega(\exp\exp(\kappa))$ running time. This is because all break-points of $d(\alpha)$ are achieved by non-randomized s.t.'s and there are already $2^{|\mathcal{Y}|^{\kappa-1}}$ $|\mathcal{Y}|$-ary trees having leaves at either depth κ or $\kappa - 1$.

In Sections IV and V we will see that, under certain conditions on $\{(X_i, Y_i)\}_{i \geq 1}$ and S, the running time of the algorithm is only *polynomial* in κ.

IV. Permutation invariant stopping times

Here we consider a special class of s.t.'s and processes $\{(X_i, Y_i)\}_{i \geq 1}$ for which the optimal tradeoff curve $d(\alpha)$ and the associated optimal s.t.'s can be computed in polynomial time in κ.

[3] We used T, $\mathcal{T}$, $\mathcal{T}_y$, and y, as possible arguments of $J_\lambda(\cdot)$. No confusion should arise from this slight abuse of notation, since all of these arguments can be interpreted as trees.

We say that a s.t. S with respect to $\{X_i\}_{i\geq 1}$ is *permutation invariant* if

$$\mathbb{P}(S \leq n|X^n = x^n) = \mathbb{P}(S \leq n|X^n = \pi(x^n))$$

for all permutations $\pi : \mathcal{X}^n \to \mathcal{X}^n$, all $x^n \in \mathcal{X}^n$ and $n \in \{1, \ldots, \kappa\}$. Examples of permutation invariant s.t.'s are $\inf\{i : X_i > c\}$ or $\inf\{i : \sum_{k=1}^{i} X_k > c\}$ for some constant $c \geq 0$ and assuming the X_i's to be positive. The notion of a permutation invariant s.t. is closely related to (and in fact slightly stronger than) that of an exchangeable s.t. as defined in [9].

We now investigate the running time in κ of the algorithm applied to permutation invariant s.t.'s. Assume S is permutation invariant and the input of the algorithm is a list of the probabilities $\mathbb{P}(S \leq n|X^n = x^n)$ for all $x^n \in \mathcal{X}^n$ and $n \in \{1, \ldots, \kappa\}$ — specifying S — and a list of $\mathbb{P}(X = x, Y = y)$ for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ — characterizing the distribution of the process $\{(X_i, Y_i)\}_{i\geq 1}$. As S is permutation invariant, we only have to specify $\mathbb{P}(S \leq n|X^n = x^n)$ for each composition (or type) of x^n. Since the number of compositions of length at most κ is upper bounded by $(\kappa + 1)^{1+|\mathcal{X}|}$ — any element $x \in \mathcal{X}$ appears at most k times in a string of length k — the list of these probabilities has only polynomial size in κ. Given x^n, the element $\mathbb{P}(S \leq n|X^n = x^n)$ in the list can be accessed in $O(\kappa)$ time.

The next theorem establishes conditions under which the algorithm worst case complexity is polynomial in κ. Its proof can be found in [1].

Theorem 1. *Let $\{(X_i, Y_i)\}_{i\geq 1}$ be i.i.d. and S be a permutation invariant s.t. with respect to $\{X_i\}_{i\geq 1}$. If all $\{T^m\}_{m=0}^{M}$ are permutation invariant, then the algorithm has a polynomial running time in κ.*

In the next section we illustrate Theorem 1 with two examples. First, we consider a TST problem that indeed can be formulated as a Bayesian change-point problem. The second example is a pure TST problem, i.e., one that cannot be formulated as a Bayesian change-point problem. For both examples we also provide an analytical solution of the Lagrange minimization problem $\min_{T\leq\kappa} J_\lambda(T)$.

V. One-step lookahead stopping times

Define

$$\mathcal{A}_n \triangleq \{y^n \in \mathcal{Y}^n : \sum_{\gamma\in\mathcal{Y}} J_\lambda(y^n\gamma) \geq J_\lambda(y^n)\},$$

and let

$$T^*_\lambda \triangleq \min\left\{\kappa, \inf\{n : Y^n \in \mathcal{A}_n\}\right\}. \tag{5}$$

In words, T^*_λ stops whenever the current cost

$$\mathbb{E}((n - S)^+|y^n) + \lambda\mathbb{P}(S > n|y^n)$$

is less than the expected cost at time $n + 1$, i.e.,

$$\mathbb{E}(((n + 1) - S)^+|y^n) + \lambda\mathbb{P}(S > n + 1|y^n)\ .$$

Recall that T^0 denotes the complete tree of depth κ and that $T(\lambda)$ denotes the minimal subtree of T whose corresponding s.t. minimizes the Lagrangian $J_\lambda(T)$. For (X_i, Y_i)'s i.i.d., Theorem 2 provides a sufficient condition on S for which $T(T^0(\lambda)) = T^*_\lambda$. In words, the s.t. T^*_λ minimizes $J_\lambda(T)$ among all s.t.'s bounded by κ, and among all stopping times minimizing $J_\lambda(T)$, the s.t. T^*_λ admits the smallest tree representation. The proof of Theorem 2 is reported in [1].

Theorem 2. *Let $\{(X_i, Y_i)\}_{i\geq 1}$ be i.i.d. If S is a s.t. with respect to $\{X_i\}_{i\geq 1}$ that satisfies*

$$\mathbb{P}(S = n|Y^{n-1}) \geq \mathbb{P}(S = n + 1|Y^n) \tag{6}$$

for all $n \in \{2, \ldots, \kappa\}$, then $T(T^0(\lambda)) = T^*_\lambda$.

Note that, unlike the algorithm, Theorem 2 provides an analytical solution only to the inner minimization problem in (4). To find the reaction delay $d(\alpha)$ one still needs to maximize over the Lagrange multipliers λ.

Using Theorems 1 and 2, we now give two examples of process $\{X_i, Y_i\}_{i\geq 1}$ and s.t. S for which the algorithm has only polynomial running time in κ.

Example 2. Let $\{(X_i, Y_i)\}_{i\geq 1}$ be i.i.d. with the X_i's taking values in $\{0, 1\}$. Consider the s.t. $S \triangleq \inf\{i : X_i = 1\}$. We have for $n \geq 2$

$$\begin{aligned}\mathbb{P}(S = n|Y^{n-1}) &= \mathbb{P}(S \geq n|Y^{n-1})\mathbb{P}(X_n = 1)\\ &\geq \mathbb{P}(S \geq n|Y^{n-1})\mathbb{P}(X_n = 0|Y_n)\mathbb{P}(X_{n+1} = 1)\\ &= \mathbb{P}(S = n + 1|Y^n)\end{aligned}$$

hence Theorem 2 yields that the one-step lookahead stopping time T^*_λ defined in (5) satisfies $T(T^0(\lambda)) = T^*_\lambda$.

We now show that the algorithm finds the set of break-points $\{\alpha_m, d_m\}_{m=0}^{M}$ and the corresponding $\{T_m\}_{m=0}^{M}$ in polynomial running time in κ. First, it can be shown that T^*_λ is permutation invariant. Since $T(T^0(\lambda)) = T^*_\lambda$ by Theorem 2, all $\{T^m\}_{m=0}^{M}$ are permutation invariant. Finally, because S is permutation invariant, applying Theorem 1 we conclude that the algorithm has indeed polynomial running time in κ.

The problem considered in this example is actually a Bayesian change-point problem, as defined at the end of Section I. Here the change-point $\Theta \triangleq S$ has distribution $\mathbb{P}(\Theta = n) = p(1 - p)^{n-1}$, where $p \triangleq \mathbb{P}(X = 1)$. The conditional distribution of Y_i given Θ is

$$\mathbb{P}(Y_i = y_i|\Theta = n) = \begin{cases}\mathbb{P}(Y_i = y_i|X_i = 0) & \text{if } i < n,\\ \mathbb{P}(Y_i = y_i|X_i = 1) & \text{if } i = n,\\ \mathbb{P}(Y_i = y_i) & \text{if } i > n.\end{cases}$$

Note that, unlike the case considered by Shiryaev (see end of Section I), the distribution of the process at the change-point differs from the ones before and after it. ◊

We now give an example that cannot be formulated as a change-point problem and for which the one-step lookahead s.t. minimizes the Lagrangian $J_\lambda(T)$.

Example 3. Let $\{(X_i, Y_i)\}_{i\geq 1}$ be i.i.d. where the X_i's and

Y_i's take value in $\{0,1\}$, and let

$$S \triangleq \inf\{i \geq 1 : \sum_{j=1}^{i} X_j = 2\}.$$

A similar computation as for Example 2 reveals that if

$$\mathbb{P}(X_i = 1|Y_i) \geq \mathbb{P}(X_i = 0|Y_i)$$

then Theorem 2 applies, showing that the one-step lookahead stopping time T^*_λ defined in (5) satisfies $T(T^0(\lambda)) = T^*_\lambda$.

Furthermore, as in the previous example, it can be shown that T^*_λ is permutation invariant. Applying Theorem 1 one deduces that the algorithm has polynomial running time in κ in this case as well.

Finally, the problem considered here is *not* a change-point problem since for $k > n$

$$\mathbb{P}(S = k|Y^n = y^n, S > n) \neq \mathbb{P}(S = k|S > n),$$

and therefore (1) does not hold. $\Diamond$

REFERENCES

[1] U. Niesen, A. Tchamkerten, and G. Wornell. Tracking stopping times (journal version). *Submitted for Publication*, December 2006.

[2] A. N. Shiryaev. *Optimal Stopping Rules*. Springer, 1978.

[3] B Yakir. Optimal detection of a change in distribution when the observations form a Markov chain with a finite state space. In *Change-point problems*, pages 346–358. Institute of Mathematical Statistics, Lecture Notes, Monograph Series, 1994.

[4] T. Z. Lai. Information bounds and quick detection of parameter changes in stochastic systems. *IEEE Transactions on Information Theory*, 44(7):2917–2929, November 1998.

[5] U. Niesen, A. Tchamkerten, and G. Wornell. Tracking stopping times. In *44th Allerton Conference on Communication, Control, and Computing*, September 2006.

[6] J. R. Baxter and R. V. Chacon. Compactness of stopping times. *Probability Theory and Related Fields*, 40(3):169–181, 1977.

[7] G. A. Edgar, A. Millet, and L. Sucheston. On compactness and optimality of stopping times. In *Martingale Theory in Harmonic Analysis and Banach Spaces*, pages 36–61. Springer, 1982.

[8] L. Breiman, J. H. Friedman, and R. Olshen. *Classification and regression trees*. Chapman & Hall, 1993.

[9] S.-Y. R. Li. Dynamic programming by exchangeability. *SIAM Journal on Computing*, 18(3):463–472, June 1998.