University of North Carolina at Chapel Hill
The University of North Carolina at Chapel Hill Department of Biostatistics Technical Report Series

Year 2009 *Paper* 11

Reinforcement Learning Design for Cancer Clinical Trials

Yufan Zhao* Michael R. Kosorok† Donglin Zeng‡

*University of North Carolina at Chapel Hill
†University of North Carolina
‡University of North Carolina

Reinforcement Learning Design for Cancer Clinical Trials

Yufan Zhao, Michael R. Kosorok, and Donglin Zeng

Abstract

We develop reinforcement learning trials for discovering individualized treatment regimens for life threatening diseases such as cancer. A temporal-difference learning method called Q-learning is utilized which involves learning an optimal policy from a single training set of finite longitudinal patient trajectories. Approximating the Q-function with time-indexed parameters can be achieved by using support vector regression or extremely randomized trees. Within this framework, we demonstrate that the procedure can extract optimal strategies directly from clinical data without relying on the identification of any accurate mathematical models, unlike approaches based on adaptive design. We show that reinforcement learning has tremendous potential in clinical research because it can select actions that improve outcomes by taking into account delayed effects even when the relationship between actions and outcomes is not fully known. To support our claims, the methodology's practical utility is illustrated in a simulation analysis. For future research, we will apply this general strategy to studying and identifying new treatments for advanced metastatic stage IIIB/IV non-small cell lung cancer, which usually includes multiple lines of chemotherapy treatment.

Statist. Med. 2009; **00**:0001–0006 *Prepared using* simauth.cls *[Version: 2002/09/18 v1.11]*

Reinforcement learning design for cancer clinical trials

Yufan Zhao, Michael R. Kosorok*,† and Donglin Zeng

Department of Biostatistics, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599, U.S.A.

SUMMARY

We develop reinforcement learning trials for discovering individualized treatment regimens for life-threatening diseases such as cancer. A temporal-difference learning method called Q-learning is utilized which involves learning an optimal policy from a single training set of finite longitudinal patient trajectories. Approximating the Q-function with time-indexed parameters can be achieved by using support vector regression or extremely randomized trees. Within this framework, we demonstrate that the procedure can extract optimal strategies directly from clinical data without relying on the identification of any accurate mathematical models, unlike approaches based on adaptive design. We show that reinforcement learning has tremendous potential in clinical research because it can select actions that improve outcomes by taking into account delayed effects even when the relationship between actions and outcomes is not fully known. To support our claims, the methodology's practical utility is illustrated in a simulation analysis. For future research, we will apply this general strategy to studying and identifying new treatments for advanced metastatic stage IIIB/IV non-small cell lung cancer, which usually includes multiple lines of chemotherapy treatment.

KEY WORDS: adaptive design; clinical trials; dynamic treatment regime; extremely randomized trees; multi-stage decision problems; non-small cell lung cancer; reinforcement learning; optimal policy; support vector regression

1. INTRODUCTION

Discovering effective therapeutic regimens for life-threatening diseases is one of the central goals of medical research. Finding powerful and general methodologies for accomplishing this discovery is a major challenge. The prevailing approach is to develop candidate therapies in the laboratory using basic science and then to test those therapies in animals and then in human clinical trials. A major problem is that very few candidate treatments make it to human clinical trials and only about 10% of treatments making it to human clinical trials demonstrate enough efficacy to be approved for marketing [1, 2]. Typical regimens for patients with certain advanced cancers (such as breast cancer, lung cancer, and ovarian cancer) utilize a single agent in combination with some platinum-based compound, and consist of multiple stages

*Correspondence to: Michael R. Kosorok, Department of Biostatistics, The University of North Carolina at Chapel Hill, Chapel Hill, NC 27599, U.S.A.
†Email: kosorok@unc.edu

Received xx xxx 2008

Revised xx xxx 2009

of treatment (especially when relapse is common). For example, many studies demonstrate that three lines of treatment can improve survival for patients with advanced non-small cell lung cancer (NSCLC). For patients who present with a good performance status and stage IIIB/IV disease, platinum-based chemotherapy is the primary treatment which can offer a modest survival advantage over best supportive care alone. Approximately 40-50% of patients in recent first-line trials received second-line treatment. Some patients who maintain a good performance status and tolerate therapy without significant toxicities will receive third-line therapy [3].

A widely used approach is to give a maximum dosage of chemotherapy drug for some period of time, followed by a period of recuperation in which no drug is given. Although this therapeutic regimen can be easily clinically implemented, this may not be the best strategy for minimizing tumor burden. Such problems have motivated the vast literature on drug-scheduling strategies. In the past few years, there has been extensive research on applications of adaptive design to clinical trials. Many investigators have developed various adaptive designs to efficiently identify clinical benefits of the treatment, and demonstrated that conducting adaptive designs can be very promising in clinical development. In general, adaptive designs for multiple courses of chemotherapy allow modification of randomization schedules based on varied probabilities of treatment assignment in order to increase the probability of success. In choosing treatments for successive courses, one of the popular adaptive designs to do this is the play-the-winner-and-drop-the-loser design, which is to repeat a treatment that is successful in a given course and otherwise switch to a different treatment. Thall *et al.* [4] provided a statistical framework for multi-course clinical trials involving some modifications of the play-the-winner-and-drop-the-loser strategy. In their proposed design, all treatments after the first course are assigned adaptively, thus increasing the amount of information available per patient. Thall *et al.* [5] presented a Bayesian adaptive design for a trial comparing two-course strategies for treating metastatic renal cancer. Each patient is fairly randomized between two treatments at enrollment, and if a patient suffers disease progression (s)he is then re-randomized among three treatments not given initially. One of the common features of these adaptive designs is the use of parametric models accounting for efficacy, toxicity, or time to some events (such as survival time). By defining a probability model, it is easy to study the design's operating characteristics under a range of parameterizations and clinical scenarios. However, as a result, it will lead to all individuals being assigned to the same level and type of treatment. Therefore, the limitation is not only to ignore the heterogeneity in treatment across individuals, but also to unsuccessfully incorporate the heterogeneity needed for optimal individualized treatment across time.

In addition to the challenge of taking into account accrued information in clinical trial designs, another major challenge is the examination of the long-term benefit of treatment due to delayed effects. If we consider the larger context of the overall therapeutic strategy, in many clinical settings a regimen with a lower initial response rate still can be the best choice in the long run. This is quite plausible due to the potential for the regimen's comparatively better delayed clinical benefit. For finding new treatment regimens with this motivation, one of the most promising approaches has been referred to variously as "dynamic treatment regimes" or "adaptive treatment strategies" [6]. In contrast with classic adaptive designs, dynamic treatment regimes can allow dosage level and type to vary with time for subject-specific needs. As a consequence, the optimal strategy is able to provide information not only on the best treatment choice from the beginning but also treatment choices that maximize outcomes for

a later time. Dynamic treatment regimes are recently emerging as a new paradigm for the treatment and long term management of chronic disease, and they have been utilized in some trials such as sequential multiple assignment randomized trials (SMART) [6] and drug and alcohol dependency studies [7]. However, to date, there are no clinical trial methodologies for discovering new treatment regimens for life-threatening diseases. Thus, for diseases like cancer, the use of clinical trials for evaluation and not discovery remains the prevailing paradigm.

In this paper, we present a general reinforcement learning framework and related statistical and computational methods for use in the clinical research arena. Reinforcement learning has been applied to treating behavioral disorders, where each patient typically has multiple opportunities to try different treatments [8]. Murphy *et al.* [9] suggest Q-learning, which is one of the most important breakthroughs in reinforcement learning, for constructing decision rules for chronic psychiatric disorders, since these chronic conditions often require sequential decision making to achieve the best clinical outcomes. Moreover, reinforcement learning has been successfully applied to the segmentation of the prostate in transrectal ultrasound images. Due to its use of knowledge obtained from the previous input image, the reinforcement learning algorithm is potentially capable of finding the appropriate local value for sub-images and extracting the prostate image [10]. However, reinforcement learning has not yet been applied to life-threatening diseases like cancer where individual patients do not have the luxury to try many different treatments. Our main aim is to illustrate the application of these methods to the discover of new treatment regimens for life-threatening diseases such as cancer. This is a paradigm shift from the standard clinical trial framework which is used for evaluating treatments but not for discovery. We consider trials in which each patient is randomized among a set of treatments at each stage and this treatment set consists of a continuous range of possibilities including, for example, a continuous range of dose levels. Therefore, rather than being constrained to a finite list of pre-specified treatments, our method allows for more general multiplicities of treatments which may include a continuum of possibilities at each stage. Reinforcement learning design has two attractive features that make it a useful tool for extracting optimal strategies directly from clinical data. First, without relying on the identification of any accurate mathematical models, it carries out treatment selection sequentially with time-dependent outcomes to determine which of several possible next treatments is best for which patients at each decision time. This feature not only helps us account for heterogeneity in treatment across individuals, but also possibly captures the best individualized therapies even when the relationship between treatments and outcomes is not fully known. Secondly, in contrast to focusing on short-term benefits, the proposed approach improves longer-term outcomes by considering delayed effect of treatments. Furthermore, we find that reinforcement learning design can extract the optimal treatment strategies while taking into account a drug's efficacy and toxicity simultaneously, which is supported by our simulation studies.

The remainder of this paper is organized as follows. In Section 2, we provide a detailed description of reinforcement learning and Q-learning. Two methods for estimating Q-functions, support vector regression (SVR) and extremely randomized trees (ERT) are presented in Section 3. The proposed "clinical reinforcement trial" method is presented in Section 4. In Section 5, we describe extensive simulation studies we conducted to discover individualized optimal treatment strategies. In Section 6, we conclude with a brief discussion that includes an important future application to NSCLC.

2. REINFORCEMENT LEARNING BACKGROUND

Our goal in this section is to introduce reinforcement learning theory, specifically, Q-learning, which will be used to discover individualized optimal therapies in cancer clinical trials.

2.1. Reinforcement learning

Over the last few decades, machine learning has become an active branch of artificial intelligence. Some of the fields studied in machine learning involve stochastic sequential decision processes, commonly referred to as reinforcement learning methods. The term "reinforcement" is subject to the occurrence of an event, in the proper relation to a response, that tends to increase the probability that the response will occur again in the same situation. From a computer science perspective, reinforcement learning is the first field to address the computational issues that arise when learning from interaction with an environment in order to achieve long-term goals. A detailed account of the history of reinforcement learning is given in Sutton and Barto [11].

The basic process of reinforcement learning involves trying a sequence of actions, recording the consequences of those actions, statistically estimating the relationship between actions and consequences, and then choosing the action that results in the most desirable consequence. In our reinforcement learning design, the thing a patient interacts with is called the "environment", which may indicate the complex system consisting of the human body and more sources of error and greater restrictions on what can be measured. While these interactions continually happen, we choose a sequence of actions applied to the patient and the environment responds to those actions and provides feedback. To be specific, we use S and A to denote random variables, where S represents the set of environmental "states" and A represents the set of possible "actions". Here "states" may represent individual patient covariates and "actions" can be denoted by various treatments or dose levels. Both variables can be discrete or continuous. Define time-dependent variables $\boldsymbol{S}_t = \{S_0, S_1, \ldots, S_t\}$, and similarly, define $\boldsymbol{A}_t = \{A_0, A_1, \ldots, A_t\}$. We also define state to possibly include past actions (i.e., S_t can include A_{t-1}). We use lower case letters, such as s and a, to denote the realized values of the random variables S and A, respectively. Also, for convenience, define $\boldsymbol{s}_t = \{s_0, s_1, \ldots, s_t\}$, and similarly, $\boldsymbol{a}_t = \{a_0, a_1, \ldots, a_t\}$. We assume the finite longitudinal trajectories are sampled at random according to a distribution P. This distribution is composed of the unknown distribution of each S_t conditional on previous $(\boldsymbol{S}_{t-1}, \boldsymbol{A}_{t-1})$. We denote these unknown conditional densities as $\{f_0, \ldots, f_T\}$, and denote expectations with respect to the distribution P as E.

As a consequence of a patient's treatment, after each time step t, the patient receives a numerical reward r_t. This could be denoted as a function (possibly random), which maps to a single number the key elements: previous state $\boldsymbol{s}_t$, action $\boldsymbol{a}_t$, and current state s_{t+1}. When $t = 0, 1, \ldots, T$, this process can be described by

$$r_t = R(\mathbf{s}_t, \mathbf{a}_t, s_{t+1}).$$

We also define $R_t = R(\mathbf{S}_t, \mathbf{A}_t, \mathbf{S}_{t+1})$. Reinforcement learning is learning what to do, how to map situations from state space S to action space A, and depending on what our goal is, how to choose a_t to maximize or minimize the expected discounted return:

$$\tilde{r}_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots + \gamma^T r_{t+T} = \sum_{k=0}^{T-t} \gamma^k r_{t+k}.$$

In this equation, γ is the discount rate ($0 \leq \gamma \leq 1$). We can interpret γ as a control to balance a patient's immediate rewards and future rewards. As γ approaches 1, we take future rewards into account more strongly. In the extreme case, where $\gamma = 1$, we fully maximize or minimize rewards over the long run.

Another key element of a reinforcement learning system is an exploration "policy", p, which maps state $\boldsymbol{s}_t$ (which can include $\boldsymbol{a}_{t-1}$) to the probability $p_t(a|\boldsymbol{s}_t)$ (the probability that action a is taken given history $\{\boldsymbol{s}_t\}$). If the policy is possibly non-stationary and non-Markovian but deterministic, we denote $\pi_t(\boldsymbol{s}_t) = a_t$. In other words, policy π_t, as a sequence of decision rules $\{\pi_1, \ldots, \pi_T\}$, is an action. Let the distribution P_π denote the distribution of training data when the policy π is used to generate actions. We can then denote expectations with respect to the distribution P_π by E_π. Let Π denote the collection of all policies. Thus the expectations E_π range over $\pi \in \Pi$. For simplicity and without loss of generality, we mainly concentrate in this paper on the goal of discovering which treatment can yield a maximized reward for a given patient. Hence seeking the policy that maximizes the expectations with respect to the sum of the rewards over the time trajectories is our ultimate goal. To accomplish this, a "value function" is established as a function of a state. Based on the condition of history $\boldsymbol{s}_t$, the value function represents the total amount of reward a patient can expect to accumulate over the future. That is,

$$V_t(\boldsymbol{s}_t) = E_\pi\Big[\sum_{k=0}^{T-t} \gamma^k R_{t+k} \mid \boldsymbol{S}_t = \boldsymbol{s}_t\Big].$$

According to this, the optimal value function can be simply defined as

$$V_t^*(\boldsymbol{s}_t) = \max_{\pi\in\Pi} V_t(\boldsymbol{s}_t) = \max_{\pi\in\Pi} E_\pi\Big[\sum_{k=0}^{T-t} \gamma^k R_{t+k} \mid \boldsymbol{S}_t = \boldsymbol{s}_t\Big].$$

Efficiently estimating the optimal value function is the most important component of almost all reinforcement learning algorithms. Since a fundamental property of value functions used throughout reinforcement learning is that they satisfy particular recursive relationships such as the Bellman equation [12], it is clear that the optimal policy, π^*, must satisfy,

$$\pi_t^*(\boldsymbol{s}_t) \in arg\max_{a_t} E\Big[R_t + \gamma V_{t+1}^*(\boldsymbol{S}_{t+1}) \mid \boldsymbol{S}_t = \boldsymbol{s}_t, \boldsymbol{A}_t = \boldsymbol{a}_t\Big].$$

Modern techniques in mathematical and computational areas have stimulated the developments of many methods for estimating the optimal value functions or optimal policies. Many of the existing methods can be categorized into one of the following two classes: dynamic programming or temporal-difference learning [11]. Bellman [12] first provided the "dynamic programming" term to show how these methods are useful to a wide range of problems. Minsky [13] first described the connection between dynamic programming and reinforcement learning. In classical dynamic programming methods, policy evaluation and policy improvement [12, 14] refer to the computation of the value function and the improved policy, respectively. The computation in both methods requires an interactive process. Combining these two methods together, we obtain two other methods called policy iteration and value iteration [15, 16]. Although dynamic programming can be applied to many types of problems, it is restricted to solving reinforcement learning problems under the Markov assumption. If this assumption is violated, dynamic programming may not be able to find an exact solution. Additionally,

dynamic programming for solving reinforcement learning problems requires knowledge of a complete and accurate model of the environment. This is almost always unrealistic in clinical settings due to the heterogeneity in the model across individual patients.

2.2. Q-learning

Although the value function plays a fundamental role in reinforcement learning, it is usually not possible to directly compute an optimal policy by just solving the Bellman optimality equation, even if we have a complete and accurate model of the environment's dynamics. Sutton [17] claims that temporal-difference (TD) learning is an alternative method to solve out optimal policies without any knowledge of the dynamic model. One fundamental expression of TD-learning is the incremental implementation, which requires less memory for estimates and less computation. Almost any TD-learning belongs to the "eligibility traces" problem. For more details on this issue, see Sutton and Barto [11] and Kaelbling *et al.* [18].

One of the most important off-policy TD-learning methods is Watkins' Q-learning [19, 20]. Q-learning no longer requires estimating the value function: it estimates a Q-function instead. Q-learning handles discounted infinite-horizon Markov decision processes (MDP). It requires no prior knowledge, is exploration insensitive and easy to implement, and is so far one of the most popular and seems to be the most effective model-free algorithm for learning from delayed reinforcement. In the setting where we don't have any information about the transition function or the probability distribution of the random variables, such a model-free method can be used to find optimal strategies from the unknown system. The motivation of Q-learning is that once the Q functions have been estimated, it is only necessary to know the state to determine the best action. From a statistical perspective, the optimal time-dependent Q-function is

$$Q_t^*(\boldsymbol{s}_t, \boldsymbol{a}_t) = E\Big[R_t + \gamma V_{t+1}^*(\boldsymbol{S}_{t+1}) \mid \boldsymbol{S}_t = \boldsymbol{s}_t, \boldsymbol{A}_t = \boldsymbol{a}_t\Big].$$

Note that since

$$V_t^*(\boldsymbol{s}_t) = \max_{a_t} Q_t^*(\boldsymbol{s}_t, a_t),$$

it is relatively easy to determine an optimal policy, which satisfies

$$\pi_t^*(\boldsymbol{s}_t) = arg\max_{a_t} Q_t^*(\boldsymbol{s}_t, a_t).$$

One-step Q-learning has the simple recursive form

$$Q_t(\boldsymbol{s}_t, \boldsymbol{a}_t) = E\left[R_t + \gamma \max_{a_{t+1}} Q_{t+1}(\boldsymbol{S}_{t+1}, a_{t+1}) \mid \boldsymbol{S}_t = \boldsymbol{s}_t, \boldsymbol{A}_t = \boldsymbol{a}_t\right]. \tag{1}$$

Under some appropriate and rigorous assumptions, Q_t has been shown to converge to Q^* with probability 1 [20]. More general convergence results were proved by Jaakkola *et al.* [21] and Tsitsiklis [22].

In learning a non-stationary non-Markovian policy with one set of finite horizon trajectories (also called a training data set)

$$\{S_0, A_0, R_0, S_1, A_1, R_1, \ldots, A_T, R_T, S_{T+1}\},$$

we denote the estimator of the optimal Q-functions based on this training data by $\widehat{Q}_t$, where $t = 0, 1, \ldots, T$. According to the recursive form of Q-learning in (1), we must estimate Q_t

backwards through time $t = T, T-1, \ldots, 1, 0$, that is, use the estimates beginning at the last time point $\widehat{Q}_T$ recursively back to $\widehat{Q}_0$ at the beginning. For convenience we set $\widehat{Q}_{T+1}$ equal to 0. In order to estimate each Q_t, we denote $Q_t(\boldsymbol{s}_t, \boldsymbol{a}_t; \theta)$ as a function of a set of parameters θ, and we allow the estimator to have different parameter sets for different time points t. Once this backwards estimation process is done, we save the sequence of $\{\widehat{Q}_0, \widehat{Q}_1, \ldots, \widehat{Q}_T\}$ for estimating optimal policies

$$\widehat{\pi}_t = arg \max_{a_t} \widehat{Q}_t(\mathbf{s}_t, \mathbf{a}_t; \theta_t),$$

where $t = 0, 1, \ldots, T$, and we thereafter use these optimal policies to test or predict for a new data set.

There are many other promising learning methods based on modifications or extensions of Q-learning, for example, Blatt, Murphy, and Zhu [23] proposed A-learning. However, some properties of these methods have not yet been carefully investigated. Due to the simple equations and minimal amount of computation, we restrict our attention in this paper to Q-learning for discovering effective therapeutic regimens in our clinical trial settings.

3. GENERAL METHODOLOGY

In this section, our main aim is to estimate the Q-function for finding the corresponding optimal policy. However, challenges may arise due to the complexity of the structure of the true Q-function, including the non-smooth maximization operation in equation (1), the high-dimension of the states variable S, the high-dimension of the action variable A, or having the action variable be continuous. In order to obtain the estimator of interest, many authors have considered different approaches in recent years. Murphy [24], Blatt *et al.* [23] and Tsitsiklis and Van Roy [25] showed that Q-learning estimation can be viewed as approximately least squares value iteration. The parameters θ_t for the t-th Q-function satisfy

$$\theta_t \in arg \min_{\theta} \mathbb{E}_n \Big[R_t + \max_{a_{t+1}} \widehat{Q}_{t+1}(\boldsymbol{S}_{t+1}, a_{t+1}; \theta_{t+1}) - Q_t(\boldsymbol{S}_t, \boldsymbol{A}_t; \theta) \Big]^2,$$

where $\mathbb{E}_n$ is the empirical expectation. This is consistent with the one-step update of Sutton and Barto [11] with $\gamma = 1$, and furthermore, it is general enough to permit function approximation and non-stationary Q-functions. Another simple and standard estimating form is provided by Murphy *et al.* [9]. They claim that Q-learning is a generalization of the familiar regression model. When the dimension of the action space is small, linear regression methods should be adequate, but in more extreme cases these methods can be questionable. Considering the discrete action space $\{a_0, a_1, \ldots, a_n\}$ with $n \geq 3$, the linear regression method will only yield as an optimal decision either a_0 or a_n due to the form of the $\max_a Q(\boldsymbol{S}, \boldsymbol{A}, a; \theta)$ term in the Q-learning implementation. Therefore, quadratic regression or higher order regression may be desired for estimating the Q-function. In this article we apply two recent flexible techniques from the machine learning literature, support vector regression (SVR) and extremely randomized trees (ERT), as our main methods to fit Q-functions and to learn an optimal policy using a training data set.

3.1. Support vector regression

The ideas underlying SVR [26] are similar but slightly different from SVM [27] within the margin-based classification scheme. The data $\boldsymbol{x}_i$ are mapped into a feature space by a nonlinear transformation Φ, which guarantees that any data set becomes arbitrarily separable as the data dimension grows [28], then a hyperplane $f(\boldsymbol{x})$ is fitted to the mapped data. One of the popular loss functions involved in SVR is known as the ϵ-insensitive loss function, which is defined as $L(f(\boldsymbol{x}_i), y_i) = (|f(\boldsymbol{x}_i) - y_i| - \epsilon)_+$, $\epsilon > 0$ [27]. Other possible loss functions include quadratic loss, Laplace loss, and Huber loss. In Q-learning, given training data $\{(\boldsymbol{x}_i, \boldsymbol{y}_i) \in X \times Y\}_{i=1}^n$, the variable X may be replaced by $\{S, A\}$ that represents states and actions information, and we define attributes $\boldsymbol{x}_{it} \in \boldsymbol{S}_t \times \boldsymbol{A}_t$, where $i = 1, \ldots, n$, $t = 0, \ldots, T$, $\boldsymbol{S}_t = \{S_0, S_1, \ldots, S_t\}$ and $\boldsymbol{A}_t = \{A_0, A_1, \ldots, A_t\}$; the variable Y may be replaced by the numerical rewards, and we assign the label index y_{it} to each total future reward value $\hat{r}_{it}$. The hyperplane $f(\boldsymbol{x})$ is equivalent to the Q function.

To fit the Q functions, let $f(\boldsymbol{x}) = \boldsymbol{w}^T\Phi(\boldsymbol{x}_i) + b$. Then SVR solves the optimization problem:

$$\min_{\boldsymbol{w},b,\xi,\xi'} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^n(\xi_i + \xi_i'),$$

$$\begin{aligned} \text{subject to} \quad & (\boldsymbol{w}^T\Phi(\boldsymbol{x}_i) + b) - y_i \le \epsilon + \xi_i, \\ & y_i - (\boldsymbol{w}^T\Phi(\boldsymbol{x}_i) + b) \le \epsilon + \xi_i', \\ & \xi_i, \xi_i' \ge 0, \ \ i = 1, \ldots, n, \end{aligned} \tag{2}$$

where ϵ is the width of the tube, ξ_i and ξ_i' are slack variables, and C is the cost of error. C is also called the tuning parameter in the machine learning field and is determined by cross validation. By minimizing the regularization term $\frac{1}{2}\|\boldsymbol{w}\|^2$ as well as the training error $C\sum_{i=1}^n(\xi_i + \xi_i')$, SVR can avoid both overfitting and underfitting the training data. The slack variables ξ_i and ξ_i' allow for some data points in the feature space to stay outside the confidence band determined by ϵ. In other words, the goal is to find a function that has at most ϵ deviation from the actually obtained targets y_i for all the training data. Errors with deviation larger than ϵ are not acceptable. A class of functions called kernels $K : \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}$ such that $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \Phi(\boldsymbol{x}_i)^T\Phi(\boldsymbol{x}_j)$ (for example, the Gaussian kernel is $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\zeta\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2)$) are used in SVR to guarantee that any data set becomes arbitrarily separable as the data dimension grows. Since the SVR function is derived within this reproducing kernel Hilbert space (RKHS) context, the explicit knowledge of both Φ and $\boldsymbol{w}$ are not needed if we have information regarding K. In this case, problem (2) is equivalent to solving an optimization dual problem equipped with Lagrange multipliers λ_i:

$$\min_{\boldsymbol{\lambda},\boldsymbol{\lambda}'} \frac{1}{2}(\boldsymbol{\lambda} - \boldsymbol{\lambda}')^T K(\boldsymbol{x}_i, \boldsymbol{x}_j)(\boldsymbol{\lambda} - \boldsymbol{\lambda}') + \epsilon\sum_{i=1}^n(\lambda_i + \lambda_i') + \sum_{i=1}^n y_i(\lambda_i - \lambda_i'),$$

$$\text{subject to} \ \ \sum_{i=1}^n(\lambda_i - \lambda_i') = 0, \ 0 \le \lambda_i, \lambda_i' \le C, \ i = 1, \ldots, n.$$

Once the above formulation is solved to get the optimal λ_i and λ_i', the approximating function at $\boldsymbol{x}$ is given by:

$$f(\boldsymbol{x}) = \sum_{i=1}^n(\lambda_i' - \lambda_i)K(\boldsymbol{x}_i, \boldsymbol{x}) + b.$$

Similar to SVM which calculates a hyperplane, the solution of an SVR function only depends on the support vectors [29]. Usually support vectors just represent a small fraction of the sample, therefore, the evaluation of the decision function is computationally efficient. This attractive property is especially useful when dealing with data sets with a low ratio of sample size to dimension. To achieve good performance by using SVR, some procedures such as data scaling, kernel and related parameter selection need to be examined very carefully. We discuss these procedures in more detail in the simulation studies below.

Compared to least-squares regression where ϵ is always zero, SVR is a more general and flexible approach for regression problems. There are several examples where SVR is successfully used in practice, and they generally perform better than other regression methods. See Chen *et al.* [30] and Smola and Scholkopf [31]. For a detailed exposition with a more computational discussion about SVR, refer to LIBSVM [32], which is a library for SVM.

3.2. Extremely randomized trees

The complex and unclear structure of the Q-function has also partly motivated the vast literature on nonparametric statistical methods and machine learning. Ernst *et al.* [33] and Geurts *et al.* [34] proposed an extremely randomized trees (ERT) method, which is called the Extra-Trees algorithm, for batch mode reinforcement learning. Unlike the classical classification and regression trees such as the Kd-tree or the pruned CART tree, this nonparametric method builds a model in the form of the average prediction of an ensemble of regression trees (called a random forest). Moreover, each tree built by this algorithm consists of strongly randomizing both attribute and cut-point choice while splitting a tree node. In addition to the number of trees G, this method depends on one parameter, called K, the maximum number of cut-direction tests at each node, and $n_{\min}$, the minimum number of elements at each leaf required to split a node. The choice of an appropriate value of G depends on the resulting compromise between computational requirements and prediction accuracy. K determines the strength of the randomization: e.g., for $K = 1$, the splits are chosen totally independent of the output variable. A larger $n_{\min}$ yields smaller trees but higher bias. The ERT algorithm builds G trees using the training data set. To determine a test at a node for each tree, this algorithm randomly selects K attributes with K randomized cut-points. A score is calculated for each test and then the one which has the highest value is kept. The algorithm stops splitting a node when the number of elements in the node is less than $n_{\min}$. The complete ERT algorithm is given in the Appendix of Geurts *et al.* [34].

Compared to standard tree-based regression methods, ERT successfully leads to significant improvements in precision. Additionally, it can dramatically decrease variance while at the same time decreasing bias, and it is very robust to outliers. ERT has been recently demonstrated in a simulation of HIV infection [35] and adaptive treatment of Epilepsy [36]. While this algorithm reveals itself to be very effective to extract a well-fitted Q from the data set, it has one drawback: the computational efficiency is relatively low especially with increasing numbers of patients in the training data set.

4. CLINICAL REINFORCEMENT TRIALS

In this section, we propose a new design and analysis method for a new kind of clinical trial for life threatening diseases, "clinical reinforcement trials". The design for these trials consists of three aspects:

First, a finite, reasonably small set of decision times is identified. These times could be either specific time points measured from trial onset or decision points in the treatment process such as the starting times of a each new line of cancer treatment. For example, in the simulation study below, we create a synthetic cancer treatment setting where patients are monitored monthly for six months and treatment for each month is determined based on patient biomarker values available at the beginning of the month. As a second example, in NSCLC, it may be more appropriate to have one decision time at the beginning of the first line of treatment, a second decision time at the beginning of the second line of treatment, and possibly a third decision time at the beginning of the third line of treatment. The third line is currently only available for certain patients and there is only one FDA approved third line treatment, and so decision possibilities are severely limited at the third decision time for this example. Note that the decision time in this instance is really a stage of treatment and not a calendar time. Other decision time sets, including hybrid variants of the previous two examples, are also possible.

Second, for each decision time, a set of possible treatments to be randomized is identified. The choice of treatments can be a continuum as mentioned earlier or a finite set and can include restrictions which may be functions of observed variables such as biomarkers. For example, in our simulations we restrict the dose of chemotherapy at the first decision time to be above a threshold so that all patients are guaranteed some initial treatment. When the set of treatments is finite, the proposed design reduces to a SMART design.

Third, a utility function is identified which can be assessed at each time point and contains an appropriately weighted combination of outcomes available at each interval between decision times and at the end of the final treatment interval. In our simulation study below, we use a combination of tumor size and overall patient health as our utility function.

We now briefly describe how a study using the proposed method would be conducted. First, a clinical reinforcement trial addressing the targeted clinical disease is designed using the principles described above. This may require developing a virtual patient model as we do in the next section. Once a design has been determined, patients are then recruited into the study and randomized to the treatment set under the protocol restrictions at each decision point, outcome measures used to compute patient state and utility are obtained, and each patient is followed through to completion of the protocol or until the end of the trial. The patient data is collected and Q-learning is applied, in combination with either SVR or ERT applied at each time point as described above, to estimate the optimal treatment rule as a function of patient variables and biomarkers, at each decision time. We allow the Q-functions to differ from decision time to decision time. This yields an individualized, time varying treatment rule that can be significantly better than the standard of care, although it may be important to validate this treatment regime in an additional phase III clinical trial. We will show in the simulation study below that our proposed approach is able to generate treatment rules that lead to improved patient outcomes. One open question which we will pursue in a later paper is how to determine sample size mathematically. Fortunately, it appears from our simulation studies that the sample sizes required are similar to and not larger than the sizes required for typical phase III trials.

5. SIMULATION STUDIES

We simulate a sequentially randomized clinical reinforcement trial as a numerical example to examine the performance of the proposed design and analysis methodology. To demonstrate that the optimal therapy found using Q-learning is superior to any other regimens, the treatments at each course are specified in terms of a continuum of dose levels of a single drug, and the comparisons we consider are between the optimal regimen identified from our proposed clinical reinforcement trial procedure and various constant-dose regimens. We first present a simple mathematical model for disease and chemotherapy which we will be using for our study. We then present the specific implementation of Q-learning which we will use for the simulation. This section concludes with a presentation of the results of the simulation study.

5.1. Simple chemotherapy mathematical model

To construct a set of training data reflecting a hypothetical cancer trial, we need a simple chemotherapy mathematical model to generate virtual patients and virtual clinical trial data. The goal for such a chemotherapy mathematical model is to allow for sufficient complexity so that the model will qualitatively generate clinically observed in vivo tumor growth patterns, while simultaneously maintaining sufficient simplicity to admit analysis. Thus, the model we present must exhibit: (1) tumor growth in the absence of chemotherapy; (2) patients' negative wellness outcomes in response to chemotherapy; (3) the drug's capability for killing tumor cells while also increasing toxicity; and (4) an interaction between tumor cells and patient wellness. To obtain data which satisfy these requirements, we propose using a system of ordinary difference equations (ODE) modeled as follows:

$$\dot{W}_t = a_1(M_t \vee M_0) + b_1(D_t - d_1),$$

$$\dot{M}_t = \Big[a_2(W_t \vee W_0) - b_2(D_t - d_2)\Big] \times \mathbf{1}\{M_t > 0\}, \tag{3}$$

where time (with month as unit) $t = 0, 1, \ldots, T-1$. Note that these changing rates yield a piecewise linear model over time. Without loss of trade-off between toxicity and efficacy, the piecewise linear model can be implemented very easily. For simplicity, we here consider tumor size instead of number of tumor cells. M_t denotes the tumor size at time t, M_0 indicates the value of tumor size when the patient is at the beginning of the study. W_t measures the negative part of wellness (toxicity), similarly, W_0 indicates the initial value of patient's wellness. D_t denotes the chemotherapy agent dose level. The value of other different parameters for the model are fixed as: $a_1 = 0.1$, $a_2 = 0.15$, $b_1 = 1.2$, $b_2 = 1.2$, $d_1 = 0.5$ and $d_2 = 0.5$. The indicator function term $\mathbf{1}\{M_t > 0\}$ in (3) represents the feature that when the tumor size is absorbed at 0, the patient has been cured, and there is no future recurrence of the tumor. Note that this model is not meant to reflect a specific cancer but to reflect a generic plausible cancer created for illustration.

Before generating simulated clinical data, it is easy to notice that the dynamic model has two state variables (W_t, M_t) and one action (treatment) variable (D_t). The state variables can be obtained via:

$$W_{t+1} = W_t + \dot{W}_t,$$

$$M_{t+1} = M_t + \dot{M}_t,$$

where $t = 0, 1, \ldots, T-1$ are the T decision times we will utilize in our simulated trial design. We generate a simulated clinical reinforcement trial with $N = 1000$ patients (replicates) with each simulated patient experiencing 6 months ($T = 6$) of treatment based on this ODE model. The initial values W_0 and M_0 for each patient are generated from independent uniform $(0, 2)$ deviates. The treatment set consists of doses of a chemotherapy agent with an acceptable dose range of $[0, 1]$, where the value 1 corresponds to the maximum acceptable dose. The values chosen for chemotherapy drug level D_0 are simulated from the uniform $(0.5, 1)$ distribution, moreover, $D_1, \ldots, D_5$ are drawn according to a uniform distribution in the interval $(0, 1)$. Thus our treatment set is restricted differently at decision time $t = 0$ than at other decision times to reflect a requirement that patients receive at least some drug at onset of treatment. Various other distribution settings for the action space are possible, and clinical researchers have tremendous flexibility in utilizing this approach.

Figure 1 provides a disease progression example of one patient to show dynamic treatment results with influence of different levels of chemotherapy agent. The system is clearly sensitive to the chemotherapy dosing regimen. Note that when the dose level switches to low, the tumor size grows to a dangerous level. Moreover, the toxicity increases (decreases) once the dosage is changed to a higher (lower) level. Note that in this model we have taken a simplistic approach for illustration and have exluded a number of potentially important factors such as the distinction between reversible and irreversible toxicities.

5.2. Q-function estimation and optimal regimen discovery

We now return to Q-learning. Utilizing the proposed ODE model, we generate a simulated clinical trial that provides a set of simulated finite horizon trajectories (or training data),

$$\{S_{0i}, A_{i0}, R_{i0}, S_{i1}, A_{i1}, R_{i1}, \ldots, A_{i5}, R_{i5}, S_{i6}\}_{i=1}^{1000},$$

where each two-dimensional state variable S_t consists of (W_t, M_t), and each continuous action variable A_t is a dose level D_t. Note that our model ignores, for simplicity, dose history in the state variables, even though past dose is technically part of the history. We wish to emphasize that it is not necessary to use all available data in Q-learning, and the user has tremendous flexibility in implementing this general approach. Continuing with our example, we will use Q-learning to maximize a sum of numerical rewards, which we now define, over six months. We assume each reward only depends on the states observed right before and after each action, that is, when $t = 0, 1, \ldots, 5$,

$$r_t = R(s_t, a_t, s_{t+1}).$$

We decompose this reward function R_t into three parts: $R_{t,1}(D_t, W_{t+1}, M_{t+1})$ due to survival status, $R_{t,2}(W_t, D_t, W_{t+1})$ due to wellness effects, and $R_{t,3}(M_t, D_t, M_{t+1})$ due to tumor size effects. It can be described by:

$$R_{t,1}(D_t, W_{t+1}, M_{t+1}) = -60, \quad \text{if patient died,}$$

otherwise,

$$R_{t,2}(W_t, D_t, W_{t+1}) = \begin{cases} 5 & \text{if } W_{t+1} - W_t \leq -0.5, \\ -5 & \text{if } W_{t+1} - W_t \geq 0.5, \\ 0 & \text{otherwise,} \end{cases}$$

$$R_{t,3}(M_t, D_t, M_{t+1}) = \begin{cases} 15 & \text{if } M_{t+1} = 0, \\ 5 & \text{if } M_{t+1} - M_t \leq -0.5, \text{ but } M_{t+1} \neq 0, \\ -5 & \text{if } M_{t+1} - M_t \geq 0.5, \\ 0 & \text{otherwise.} \end{cases}$$

In most phase III clinical trials, the primary endpoint of clinical interest is the overall survival (OS): this is why we put -60 as a high penalty for patient's death. Additionally, we assign a relative high value 15 as a bonus when a patient is cured.

We assume that survival status depends on both toxicity and tumor size. For each time interval $(t-1, t], t = 1, \ldots, 6$, we define the hazard function as $\lambda(t)$, which satisfies

$$log\lambda(t) = \mu_0 + \mu_1 W_t + \mu_2 M_t,$$

where μ_0, μ_1, and μ_2 are constant pre-specified parameters. In particular, assigning $\mu_1 = \mu_2 = 1$ indicates that we consider wellness and tumor size to have an equally weighted influence on the survival rate. The survival function is then

$$\Delta F(t) = \exp\left[-\Delta\Lambda(t)\right],$$

where $\Delta\Lambda(t) = \int_{t-1}^{t} \lambda(s)d(s)$ is the cumulative hazard function. The reason the term $R_{t,1}(D_t, W_{t+1}, M_{t+1})$ is expressed as a function of W_{t+1} and M_{t+1} is that the hazard function is only determined by the states at the end of each time interval. The conditional probability of death for each time interval is $p = 1 - \Delta F(t)$. The survival status (with death coded as 1) is drawn according to a Bernoulli distribution $B(p)$. Overall, by letting $\gamma = 1$ (we would like to fully consider maximizing rewards in the long run), the one-step Q-learning with recursive form is utilized, with $Q_t(S_t, A_t)$ predicting

$$\widehat{R}_t = R_t + \max_{a_{t+1}} \widehat{Q}_{t+1}(S_{t+1}, a_{t+1}),$$

where $R_t = R_{t,1}(D_t, W_{t+1}, M_{t+1}) + R_{t,2}(W_t, D_t, W_{t+1}) + R_{t,3}(M_t, D_t, M_{t+1}), t = 0, \ldots, 5$. Note that $\hat{r}_t$ mentioned previously is defined as a realization of $\widehat{R}_t$. This recursive estimation process is called SARSA (state, action, reward, next state, next action) in the reinforcement learning literature.

To obtain the estimator $\widehat{Q}_t$, we apply SVR and ERT respectively for fitting Q_t backward, and save the results as $\{\widehat{Q}_5, \widehat{Q}_4, \ldots, \widehat{Q}_0\}$. Figure 2 illustrates the treatment plan and relevant Q-function estimation procedures. Because of the inner product property of the kernel in SVM/SVR, scaling the data before applying SVR is very important. Another advantage for scaling is to avoid states with greater numeric ranges dominating those with smaller numeric ranges. In our simulation studies, every variable is scaled to zero mean and unit variance, and the center and scale values are saved and used for later predictions. To do fitting of $\widehat{Q}_t$ via SVR, we select the Gaussian kernel (or Radial Basis Function), $K(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\zeta\|\boldsymbol{x} - \boldsymbol{y}\|^2\right)$, because the Gaussian kernel can nonlinearly map samples into a higher dimensional space. Consequently, it can handle the case when the relation between rewards (labels) and states and actions (attributes) is nonlinear. In the SVR approach there are two hyperparameters involved with the Gaussian kernel: ζ and the tuning parameter C. To maximize the performance of the proposed method, we apply a grid search to choose C and ζ by using cross-validation. Trying exponentially growing sequences of C and ζ is recommended as a practical method

to identify good hyperparameters. Specifically, for each t in our simulated example, given a straightforward coarse grid search with $C = 2^{-5}, 2^{-3}, \ldots, 2^{15}$ and $\zeta = 2^{-15}, 2^{-13}, \ldots, 2^{3}$, we apply cross-validation to each candidate pair (C, ζ), and then select the pair that yields the highest cross-validation rate. To do fitting $\widehat{Q}_t$ via ERT, we need to be careful with the choice of parameters G, K and $n_{\min}$. Based on empirical studies, Geurts *et al.* (2006) suggest that the default value of K should be equal to the number of attributes in the regression problem. Thus we fix K as the dimension of state variables plus the dimension of action variables, which is equal to 3 in our case. To maintain good precision and small bias, G and $n_{\min}$ have been chosen equal to 50 and 2, respectively.

Based on the sequential estimators $\{\widehat{Q}_5, \widehat{Q}_4, \ldots, \widehat{Q}_0\}$, the individualized optimal policies as a function of the state variables are estimated by maximizing over dose level (i.e., a_t):

$$\widehat{\pi}_t(s_t) = arg \max_{a_t} \widehat{Q}_t(s_t, a_t; \hat{\theta}_t).$$

In order to evaluate how the above estimated treatment policies performed, we generated a virtual phase III clinical trial with 200 patients per each of 11 treatments consisting of the estimated optimal treatment regime and each of the 10 possible fixed dose levels ranging from 0.1 to 1.0 with increments of size 0.1. The initial values of W_0 and M_0 for the patients were randomly chosen from the same uniform distribution used in the training data.

The entire algorithm for Q-function estimation and optimal regimen evaluation is summarized as follows:

1. Inputs: a set of training data consists of attributes $\boldsymbol{x}$ (states s_t, actions a_t) and index $\boldsymbol{y}$ (rewards r_t), i.e. $\{(s_t, a_t, r_t)_i, t = 0, \ldots, T, i = 1, \ldots, N\}$.
2. Initialization: Let $t = T + 1$ and $\widehat{Q}_{T+1}$ be a function equal to zero on $\boldsymbol{S}_t \times \boldsymbol{A}_t$.
3. Iterations: repeat computations until stopping conditions are reached ($t = 0$).
   (a) $t \leftarrow t - 1$.
   (b) Q_t is fitted with support vector regression (SVR) or extremely randomized trees (ERT) through the following recursive equation:
   $$Q_t(s_t, a_t) = r_t + \max_{a_{t+1}} \widehat{Q}_{t+1}(s_{t+1}, a_{t+1}) + error.$$
   (c) Use cross-validation to choose tuning parameters C and ζ if fitting Q_t via SVR with Gaussian kernel; choose plausible values of parameters K, G, n_{min} if fitting Q_t via ERT ($K = 3, G = 50, n_{min} = 2$ in our simulation).
4. Given the sequential estimates of $\{\widehat{Q}_0, \widehat{Q}_1, \ldots, \widehat{Q}_5\}$, the sequential individualized optimal polices $\{\widehat{\pi}_0, \ldots, \widehat{\pi}_5\}$ for application to the virtual phase III trial are computed.

5.3. Simulation results

In the simulated phase III trial, we examine W_t, M_t and the patients' cumulative survival probability. All of these quantities were averaged over the 200 virtual patients. These averages at time $t = 6$ are given, along with the survival results, in Table 1.

We used a sample size of 1000 for our simulated clinical reinforcement trial and estimated the optimal treatment policy using both SVR and ERT. For the sake of simplicity, unless stated explicitly otherwise, we only show results in the figures for the SVR method, since we

obtain very similar results when we estimate optimal therapy using ERT. In Figures 3 and 4, trajectories (wellness and tumor size, respectively) that would have been observed by putting the patients on constant-dose regimens have been plotted. Note that the wellness measure has been inverted so that larger values represent worse health. This is to make comparisons with tumor size more direct. We test the behavior of estimated optimal regimens on 200 new simulated patients by comparing the outcomes using $\widehat{\pi}_t$ from the $\widehat{Q}_t$ $(t = 0, \ldots, 5)$ against the results obtained using 10 different fixed D_t levels $(t = 0, \ldots, 5)$ in the ODE model. As shown in both Figure 3 and Figure 4, the optimal regimens derived from Q-learning do not have better performance compared to some constant dosing regimens. This is not beyond our expectation. Because when a higher dose level decreases tumor size, it can yield a higher toxicity simultaneously, and vice versa. However, due to our reward functions structure, the estimated optimal policies have an appealing feature that seeks a good balance between toxicity and efficacy. Figure 5 illustrates that the estimated optimal regimen is absolutely superior to any constant-dose regimen when we combine toxicity and efficacy $(W_t + M_t)$ into one comparison criterion. Table 1 agrees with this conclusion by respectively presenting $W_6 + M_6 = 3.269$ (SVR) or $W_6 + M_6 = 3.194$ (ERT) as the lowest number compared to the others. Most notably, although the regimen derived from simulated data shows suboptimal results in the first three months, it achieves the best performance eventually. These findings agree well with reinforcement learning's substantially powerful long-run capabilities.

Figure 6 provides the dynamic optimal regimen for an individual patient as well as the toxicity and efficacy values during the whole trial. This simulated patient comes into the trial with initial condition $W_0 = 0.30$ and $M_0 = 1.05$. Optimal therapy begins with a very high dose $D_0 = 1.00$ aimed at reducing the patient's tumor burden. The patient is then monitored for the following month and then treated with another two consecutive high doses ($D_1 = 0.74$, $D_2 = 1.00$). In the third month, the tumor size suddenly reaches 0, i.e. the patient has been cured. As expected, we find that the dosage to be administrated rapidly reduces to 0 in the following months. Patients who recover after three months will not receive high dosing anymore because the high dose will likely result in unnecessarily high toxicity. As we can see, rather than the constant dose level for each t, optimal therapy usually has an up-and-down structure due to its adaptive properties. This is an important result which demonstrates that the optimal policy can be approximated very well by reinforcement learning.

Finally, compared to all fixed-level doses, Table 1 clearly shows that the therapy found using the Q-learning approach with either SVR or ERT has better performance in terms of cumulative survival probability (CSP). Both SVR and ERT appear to perform equally well with comparable computational burden.

6. DISCUSSION

We have developed a reinforcement learning method for discovering effective therapeutic regimens in clinical trial design. To investigate the validity of such a purely data (model-free) driven approach, we have generated clinical data by relying on a set of hypothetical (and simplistic) but plausible ODE models. Based on these simulated data, we have found that reinforcement learning is indeed able to identify individualized optimal regimens in clinical trials which consist of multiple courses of treatment. Such regimens can reduce tumor burden while taking into account a drug's toxicity. Treatment delay effects, which is an important issue

that must be considered for longer term outcomes, are fully assessed by this method. Another appealing feature of our approach is the incorporation of Q-learning methodology with SVR and ERT. Hence even in a data set comprised of high-dimensional attributes, our method is capable of obtaining promising results without much computational burden.

There are a number of challenges we expect to face in future research. First of all, in our study we have defined the reward as a straightforward function to map states and actions into some integer numbers (15, 5, 0, -5 and -60). This simplistic reward function construction along with Q-learning represents an attractive way for trading off efficacy against toxicity and death. However, it is unclear how changing these numbers affects the resulting optimal regimens identified during discovery of effective therapeutic strategies. Understanding the robustness of Q-learning to numerical reward choices is an interesting problem and clearly deserves further investigation. Secondly, since the choice of reward function plays a crucial role in reinforcement learning, therefore, it is very important to consider alternative rewards directly reflecting primary endpoints (such as overall survival, progression-free survival, quality-of-life adjusted survival, freedom from side effects, etc.) in clinical trial designs. One of many feasible approaches for accomplishing this is to perform retrospective analyses on existing clinical trial data to identify clinical factors that influence the outcome of patients treated with chemotherapy drugs, and to build a model that can be used in practice to predict long-term survival in this patient population. Such a model may assist us in building a more plausible reward function, and thereafter determining a regimen which is as close as possible to an optimal policy. To construct a clinically relevant reward function, we believe that close collaboration with clinical researchers is required. An interesting illustrative example of a related strategy is shown by Ernst *et al.* [35]. They consider discounted instantaneous costs (which is a continuous function directly associated with actions) as their reward function: the rationale behind this comes from a validated and identified HIV model [37].

In this paper, we observed that with sample size $N = 1000$ for a clinical reinforcement trial, using SVR or ERT leads to a reasonably low bias for estimating optimal regimens. The evidence for this is the confirmed success of the discovered treatment regimen on an independent sample of 200 simulated patients. Clearly, in many settings, this assumption may be violated due to the complexity associated with the performance of the approximation on the Q function, the high-dimensional state or action space, the horizon time T, the connection with SVR or ERT, and more importantly, estimation accuracy. Therefore, an interesting but potentially difficult question would be: how to determine the number of patients N required in the clinical reinforcement trial, which allows utilizing the SVR or ERT to fit Q and can be guaranteed to obtain a regimen that is very close to the optimal one. All these theoretical issues are under investigation and will be presented elsewhere.

Since the work of this article is motivated by the clinical question of proper treatment for Stage IIIB/IV NSCLC, as examined by several clinical trials conducted at the UNC Lineberger Comprehensive Cancer Center (LCCC), an important future application is to refine our model to more accurately reflect NSCLC and the associated treatment issues. The goal of this future study is to compare strategies for multiple lines of treatment for patients with advanced NSCLC who have not been treated previously with systemic therapy. In our future study we will apply reinforcement learning to discover individualized optimal regimens while restricting attention to first-line and second-line only, since there is only one approved agent (Erlotinib) indicated for third-line treatment [38].

First-line treatment primarily consists of platinum-based doublets that include cisplatin,

gemcitabine, pemetrexed, paclitaxel, carboplatin or vinorelbine. Numerous studies have compared these various platinum doublets, the great majority of these trials have concluded that all such regimens are comparable in their clinical efficacy. As an example, see Scagliotti *et al.* [39]. Their study represents the largest number of patients entered into a single phase III study using either cisplatin + gemcitabine or cisplatin + pemetrexed regimen. Noninferiority was demonstrated because the median survival time was an identical 10.3 months in each arm. In addition to platinum-based doublets, some phase III studies have examined the efficacy of various targeted therapies. Bevacizumab plus paclitaxel + carboplatin in the treatment of selected patients with NSCLC showed a significant survival benefit [40], however, with the risk of increased treatment-related deaths. Cetuximab plus cisplatin + vinorelbine demonstrated superior survival in patients with advanced EGFR-detectable NSCLC [41]. The strategies of first-line therapy are essentially based on these four targeted combination therapies, the choice depends on a number of factors, including the patient's histology type, toxicity profile, smoking history, VEGF level, EGFR expression and race [39, 40, 41].

There are three agents approved for treating patients in a second-line regimen: docetaxel, pemetrexed, and erlotinib. Similar to the first-line regimen, these agents appear to have similar efficacies in terms of response and overall survival, but have significantly different toxicity profiles. The choice of agent also depends on many factors, including the patient's number of prior regimens, response to prior chemotherapy, the risk for neutropenia, EGFR expression, and patient preference [42, 43, 44, 45].

Due to the complexity of the biomarkers and unclear toxicities, the trial described here was motivated by the desire to compare those agents in a randomized fashion, and the belief that different combinations given between first-line and second-line may have interactive effects. Another very important factor omitted from the cancer treatment model given in this paper is the potentially enormous difference between reversible and irreversible toxicity. This difference will need to be incorporated in future studies on this topic. Despite the difficulty of discovering the superior therapies, another primary challenge is to determine the optimal second-line regimen's starting time, either immediate or delayed after induction therapy, having the highest overall survival probability. Although Fidias *et al.* [44] provided some results to suggest that an immediate transition to second-line therapy may be optimal, whether these findings are specific to docetaxel or would hold true for other common second-line options such as pemetrexed or erlotinib remains unknown. Hence, optimal timing of second-line therapy with a non-cross-resistant agent is an important issue and still remains unclear. Furthermore, including biomarkers as covariates to assess possible effects, such as biomarker treatment interactions and biomarker second-line-timing interactions, is potentially useful. In our future research, a better understanding of prognostic factors is needed, this may lead us to discover better individualized therapies using reinforcement learning.

ACKNOWLEDGEMENTS

The authors would like to thank Dr. Mark Socinski for helpful discussions on non-small cell lung cancer treatment and the Reinforcement Learning Group at the University of North Carolina for many stimulating exchanges. The research was funded in part by grant CA075142 from the U.S. National Cancer Institute and from pilot funding provided by the Center for Innovative Clinical Trials at the UNC Gillings School of Global Public Health. The authors also thank two anonymous referees who provided helpful comments that led to a significantly improved paper.

Prepared using simauth.cls

REFERENCES

1. Hogberg T. Widening bottlenecks in drug discovery Glimpses from Drug Discovery Technology Europe. *Drug Discovery Today* 2005; **10**(12):820–822.
2. Food and Drug Administration. Innovation or stagnation: challenge and opportunity on the critical path to new medical products. *White Paper*, 2004.
3. Stinchcombe TE, Socinski MA. Considerations for second-line therapy of non-small cell lung cancer. *The Oncologist* 2008; **13**(suppl 1):28–36.
4. Thall PF, Millikan RE, Sung HG. Evaluating multiple treatment courses in clinical trials. *Statistics in Medicine* 2000; **19**:1011–1028.
5. Thall PF, Wooten LH, Logothetis CJ, Millikan RE, Tannir NM. Bayesian and frequentist two-stage treatment strategies based on sequential failure times subject to interval censoring. *Statistics in Medicine* 2007; **26**:4687–4702.
6. Murphy SA. An experimental design for the development of adaptive treatment strategies. *Statistics in Medicine* 2005; **24**:1455–1481.
7. Murphy SA, Lynch KG, Oslin D, McKay JR, TenHave T. Developing adaptive treatment strategies in substance abuse research. *Drug and Alcohol Dependence* 2007; **88S**:S24–S30.
8. Pineau J, Bellemare MG, Rush AJ, Ghizaru A, Murphy SA. Constructing evidence-based treatment strategies using methods from computer science. *Drug and Alcohol Dependence* 2007; **88S**:S52–S60.
9. Murphy SA, Oslin DW, Rush AJ, Zhu J. Methodological challenges in constructing effective treatment sequences for chronic psychiatric disorders. *Neuropsychoarmacology* 2007; **32**:257–262.
10. Sahba F, Tizhoosh HR, Salama MM. Application of reinforcement learning for segmentation of transrectal ultrasound images. *BMC Medical Imaging* 2008; **8**(8).
11. Sutton RS, Barto AG. *Reinforcement Learning: An Introduction.* MIT Press: Cambridge, MA, 1998.
12. Bellman RE. *Dynamic Programming.* Princeton University Press: Princeton, 1957.
13. Minsky ML. Steps toward artificial intelligence. *Proceedings of the Institute of Radio Engineers* 1961; **49**:8–30.
14. Howard R. *Dynamic Programming and Markov Processes.* MIT Press, 1960.
15. Puterman ML, Shin MC. Modified policy iteration algorithms for discounted Markov decision problems. *Management Science* 1978; **24**:1127–1137.
16. Bertsekas DP. *Dynamic Programming: Deterministic and Stochastic Models.* Prentice-Hall: Englewood Cliffs, 1987.
17. Sutton RS. Learning to predict by the method of temporal differences. *Machine Learning* 1988; **3**(1):9–44.
18. Kaelbling LP, Littman ML, Moore AW. Reinforcement learning: a survey. *Journal of Artificial Intelligence Research* 1996; **4**:237–285.
19. Watkins CJCH. *Learning From Delayed Rewards.* Ph.D. Thesis, King's College, Cambridge, UK, 1989.
20. Watkins CJCH, Dayan P. Q-learning. *Machine Learning* 1992; **8**:279–292.
21. Jaakkola T, Jordan MI, Singh SP. On the convergence of stochastic iterative dynamic programming algorithms. *Neural Computation* 1994; **6**:1185–1201.
22. Tsitsiklis JN. Asynchronous stochastic approximation and Q-learning. *Machine Learning* 1994; **16**:185–202.
23. Blatt D, Murphy SA, Zhu J. *A*-learning for approximate planning. *Unpublished Manuscript*, 2004.
24. Murphy SA. A generalization error for Q-learning. *Journal of Machine Learning Research* 2005; **6**:1073–1097.
25. Tsitsiklis JN, Van Roy B. Feature-based methods for large scale dynamic programming. *Machine Learning* 1996; **22**:59–94.
26. Vapnik V, Golowich S, Smola A. Support vector method for function approximation, regression estimation, and signal processing. *Advances in Neural Information Processing Systems* 1997; **9**:281–287.
27. Vapnik V. *The Nature of Statistical Learning Theory.* Springer: New York, 1995.
28. Cover TM. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers* 1965; **14**:326–334.
29. Cortes C, Vapnik V. Support-vector networks. *Machine Learning* 1995; **20**:273–297.
30. Chen BJ, Chang MW, Lin CJ. Load forecasting using support vector machines: a study on EUNITE competition 2001. *IEEE Transactions on Power Systems* 2004; **19**:1821–1830.
31. Smola A, Scholkopf B. A tutorial on support vector regression. *Statistics and Computing* 2004; **14**:199–222.
32. LIBSVM: a library for support vector machines (version 2.31). http://www.csie.ntu.edu.tw/~cjlin/libsvm [7 September 2001].
33. Ernst D, Geurts P, Wehenkel L. Tree-based batch model reinforcement learning. *Journal of Machine Learning Research* 2005; **6**:503–556.
34. Geurts P, Ernst D, Wehenkel L. Extremely randomized trees. *Machine Learning* 2006; **11**:3–42.
35. Ernst D, Stan GB, Goncalves J, Wehenkel L. Clinical data based optimal STI strategies for HIV: a reinforcement learning approach. *Proceedings of the* 45^{th} *IEEE Conference on Decision and Control* 2006.

36. Guez A, Vincent R, Avoli M, Pineau J. Adaptive treatment of Epilepsy via batch-mode reinforcement learning. *Innovative Applications of Artificial Intelligence* 2008.
37. Adams B, Banks H, Kwon HD, Tran H. Dynamic multidrug therapies for HIV: Optimal and STI control approaches. *Mathematical Biosciences and Engineering* 2004; **1**:223–241.
38. Shepherd FA, Pereira JR, Ciuleanu T, Tan EH, Hirsh V, Thongprasert S, Campos D, Maoleekoonpiroj S, Smylie M, Martins R, van Kooten M, Dediu M, Findlay B, Tu D, Johnston D, Bezjak A, Clark G, Santabarbara P, Seymo L. Erlotinib in previously treated non-small-cell lung cancer. *The New England Journal of Medicine* 2005; **353**:123–132.
39. Scagliotti GV, Parikh P, Von Pawel J, Biesma B, Vansteenkiste J, Manegold C, Serwatowski P, Gatzemeier U, Digumarti R, Zukin M, Lee JS, Mellemgaard A, Park K, Patil S, Rolski J, Goksel T, de Marinis F, Simms L, Sugarman KP, Gandara D. Phase III study comparing cisplatin plus gemcitabine with cisplatin plus pemetrexed in chemotherapy-naive patients with advanced-stage non-small-cell lung cancer. *Journal of Clinical Oncology* 2008; **26**.
40. Sandler A, Gray R, Perry MC, Brahmer J, Schiller JH, Dowlati A, Lilenbaum R, Johnson DH. Paclitaxel-Carboplatin alone or with bevacizumab for non-small-cell lung cancer. *The New England Journal of Medicine* 2006; **355**:2542–2550.
41. Pirker R, Szczesna A, Von Pawel J, Krzakowski M, Ramlau R, Park K, Gatzemeier U, Bajeta E, Emig M, Pereira JR. FLEX: A randomized, multicenter, phase III study of cetuximab in combination with cisplatin/vinorelbine (CV) versus CV alone in the first-line treatment of patients with advanced non-small cell lung cancer (NSCLC). *Journal of Clinical Oncology* 2008; May 20 suppl, abstr 3.
42. Shepherd FA, Dancey J, Ramlau R, Mattson K, Gralla R, O'Rourke M, Levitan N, Gressot L, Vincent M, Burkes R, Coughlin S, Kim Y, Berille J. Prospective randomized trial of docetaxel versus best supportive care in patients with non-small-cell lung cancer previously treated with platinum-based chemotherapy. *Journal of Clinical Oncology* 2000; **18**:2095–2103.
43. Hanna N, Shepherd FA, Fossella FV, Pereira JR, de Marinis F, Von Pawel J, Gatzemeier U, Chang T, Tsao Y, Pless M, Muller T, Lim HL, Desch C, Szondy K, Gervais R, Shaharyar, Manegold C, Paul S, Paoletti P, Einhorn L, Bunn PA. Randomized phase III trial of pemetrexed versus docetaxel in patients with non-small-cell lung cancer previously treated with chemotherapy. *Journal of Clinical Oncology* 2004; **22**:1589–1597.
44. Fidias P, Dakhil S, Lyss A, Loesch D, Waterhouse D, Cunneen J, Chen R, Treat J, Obasaju C, Schiller J. Phase III study of immediate versus delayed docetaxel after induction therapy with gemcitabine plus carboplatin in advanced non-small-cell lung cancer: Updated report with survival. *ASCO* 2007; **25**:June 20 suppl, LBA7516.
45. Ciuleanu TE, Brodowicz T, Belani CP, Kim J, Krzakowski M, Laack E, Wu Y, Peterson P, Adachi S, Zielinski CC. Maintenance pemetrexed plus best supportive care (BSC) versus placebo plus BSC: A phase III study. *ASCO* 2008; May 20 suppl, abstr 8011.

Prepared using simauth.cls

Table I. Summary of main simulation results

	Optimal Regimens		Constant-Dose Regimens									
	RL^1	RL^2	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
W_6	**1.276**	**1.328**	-0.411	0.129	0.669	1.217	1.783	2.375	3.016	3.705	4.421	5.141
M_6	**1.993**	**1.866**	4.737	4.017	0.300	2.654	2.133	1.658	1.203	0.812	0.496	0.257
$W_6 + M_6$	**3.269**	**3.194**	4.326	4.146	3.970	3.870	3.916	4.033	4.219	4.517	4.917	5.397
CSP	**0.442**	**0.441**	0.240	0.292	0.345	0.377	0.363	0.331	0.275	0.189	0.061	0.003

Note: Results such as the wellness (W_6), tumor size (M_6), wellness plus tumor size ($W_6 + M_6$) and cumulative survival probability (CSP) are given for the optimal regimen using reinforcement learning methods via SVR (RL^1) and ERT (RL^2), and for the constant-dose regimen which ranges from 0.1 to 1.0. All numbers are averaged over 200 patients. Entries for superior strategies are given in boldface type.

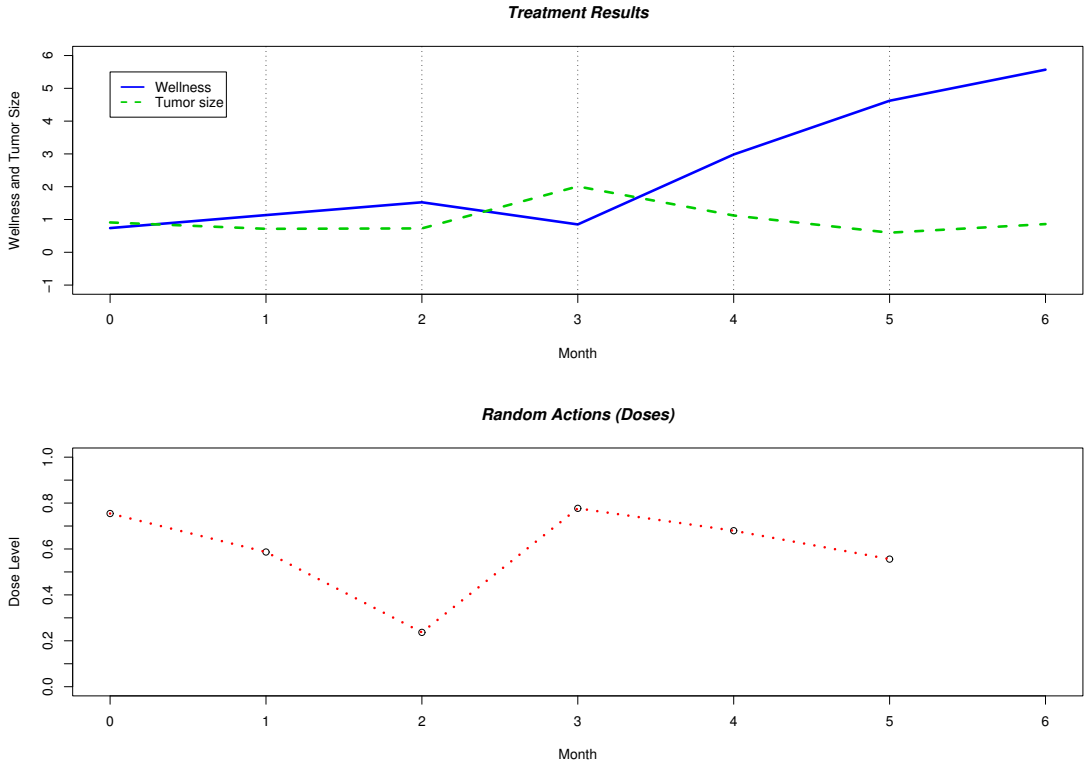

Figure 1. Representation of the disease progression for a patient treated from randomized chemotherapy drug. The solid curve represents the negative part of patient's wellness, the dashed curve represents the tumor size, and the dotted curve represents the randomized treatment.

Prepared using simauth.cls

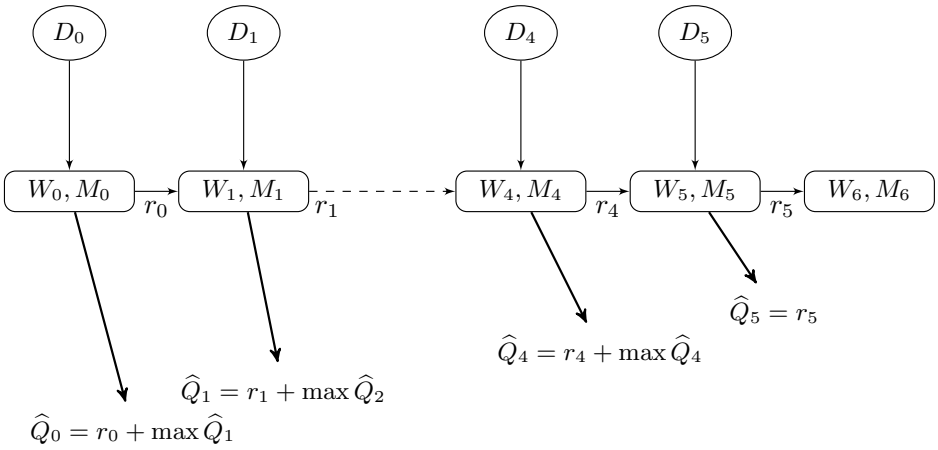

Figure 2. Treatment plan and the procedure for obtaining the sequential estimator $\{\widehat{Q}_5, \widehat{Q}_4, \ldots, \widehat{Q}_0\}$.

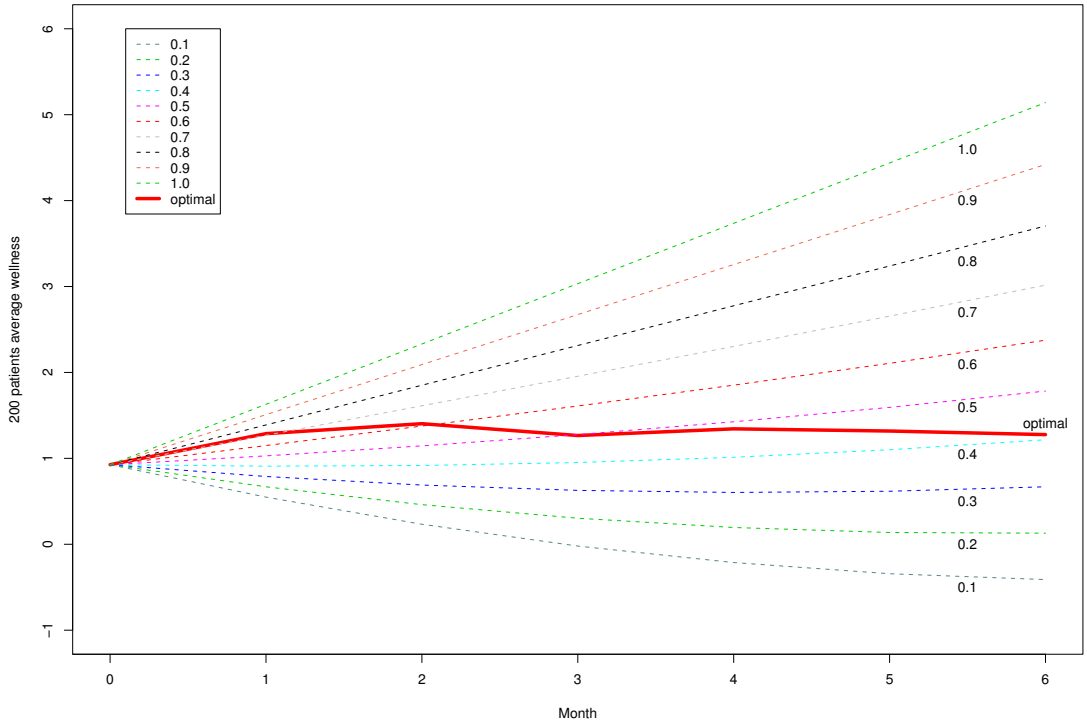

Figure 3. Plots of averaged value of "wellness" (negative part) for 10 different constant-dose regimens compared to optimal regimen. The results are based on 200 patients. Dashed curves represent the constant-dose regimens, and a solid curve represents the optimal regimen.

Prepared using simauth.cls

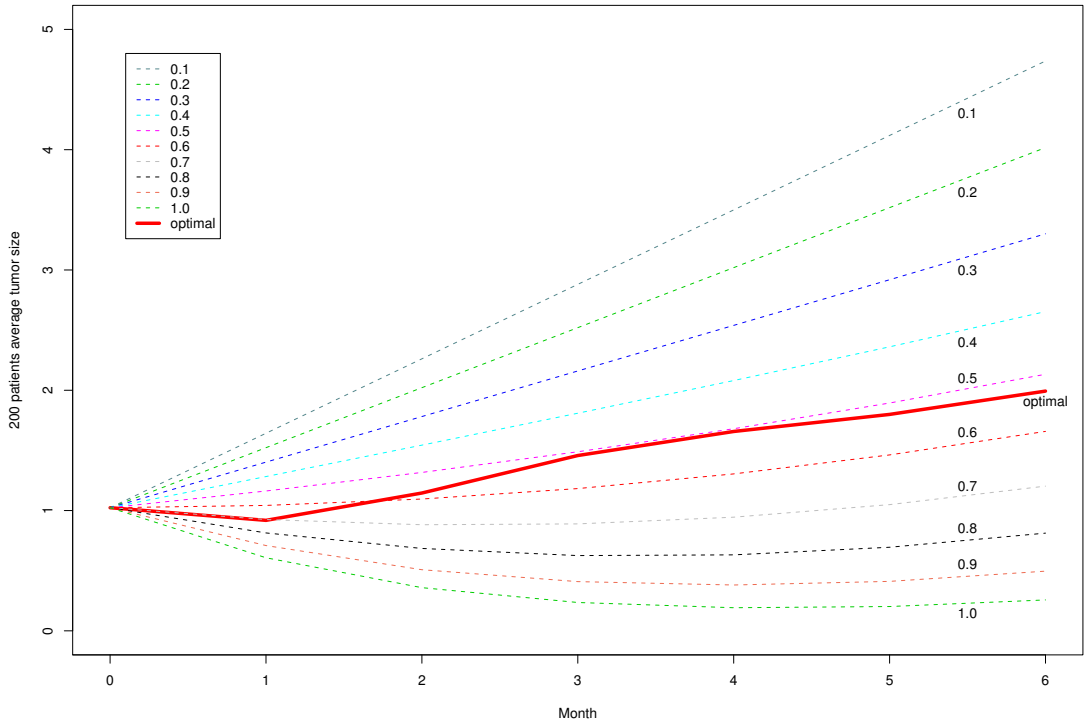

Figure 4. Plots of averaged value of "tumor size" for 10 different constant-dose regimens compared to the optimal regimen. The results are based on 200 patients. Dashed curves represent the constant-dose regimens, and a solid curve represents the optimal regimen.

Prepared using simauth.cls

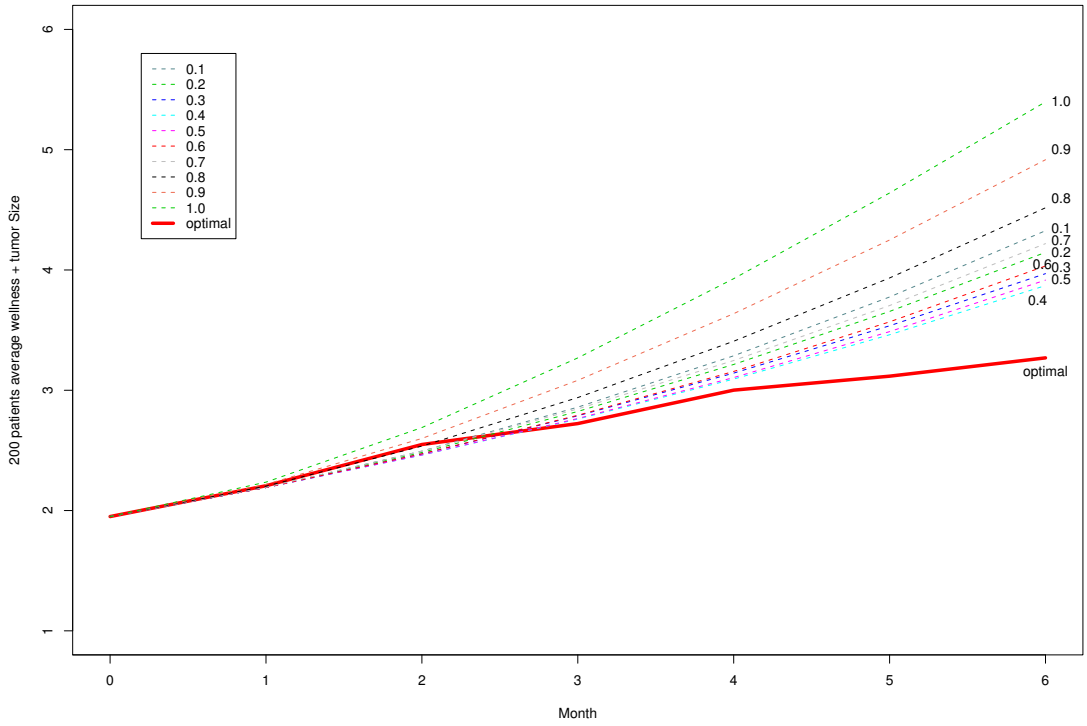

Figure 5. Plots of averaged value of "wellness + tumor size" for 10 different constant-dose regimens compared to optimal regimen. The results are based on 200 patients. Dashed curves represent the constant-dose regimens, and a solid curve represents the optimal regimen.

Prepared using simauth.cls

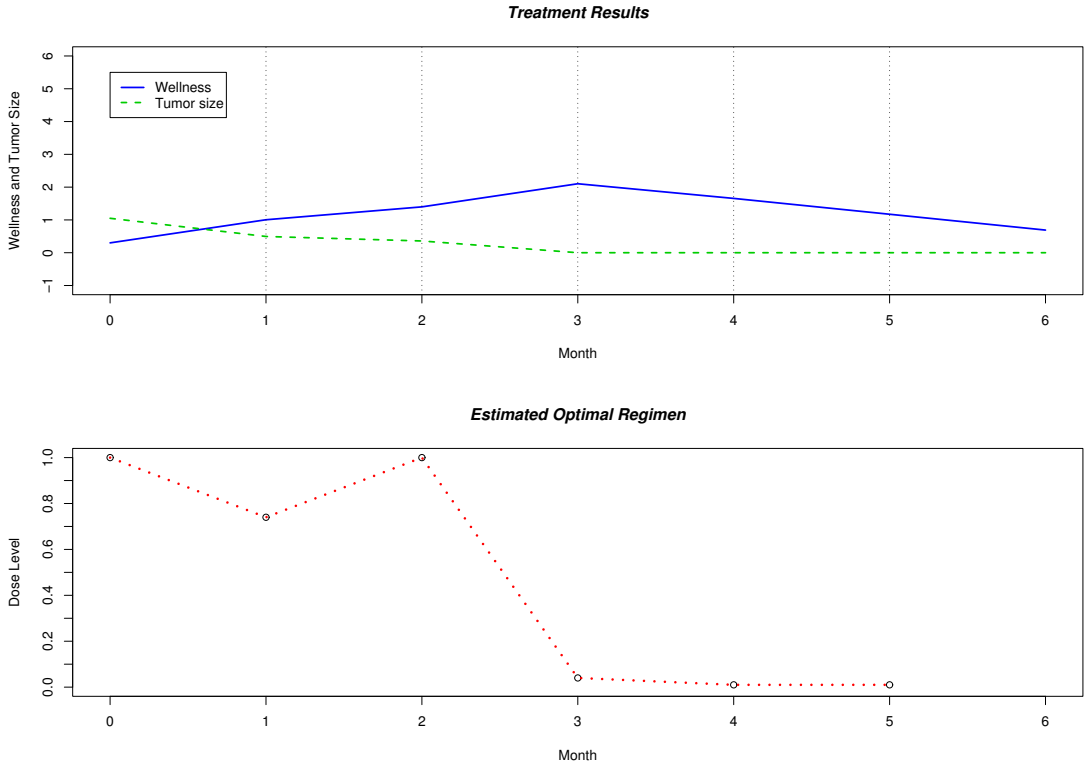

Figure 6. Representation of the optimal treatment for a patient with $W_0 = 0.30$ and $M_0 = 1.05$. The optimal treatment sequence ($D_t \in \{1.00, 0.74, 1.00, 0.04, 0.01, 0.01\}$) is computed by the proposed reinforcement learning methods on clinical data generated by 1000 patients. The solid curve represents the negative part of patient's wellness, the dashed curve represents the tumor size, and the dotted curve represents the estimated optimal regimen.