Adaptive Treatment of Epilepsy via Batch-mode Reinforcement Learning

Arthur Guez, Robert D. Vincent
School of Computer Science
McGill University
Montreal, Quebec Canada

Massimo Avoli
Montreal Neurological Institute
McGill University
Montreal, Quebec Canada

Joelle Pineau
School of Computer Science
McGill University
Montreal, Quebec Canada

Abstract

This paper highlights the crucial role that modern machine learning techniques can play in the optimization of treatment strategies for patients with chronic disorders. In particular, we focus on the task of optimizing a deep-brain stimulation strategy for the treatment of epilepsy. The challenge is to choose which stimulation action to apply, as a function of the observed EEG signal, so as to minimize the frequency and duration of seizures. We apply recent techniques from the reinforcement learning literature—namely fitted Q-iteration and extremely randomized trees—to learn an optimal stimulation policy using labeled training data from animal brain tissues. Our results show that these methods are an effective means of reducing the incidence of seizures, while also minimizing the amount of stimulation applied. If these results carry over to the human model of epilepsy, the impact for patients will be substantial.

Introduction

Clinicians treating individuals with chronic disorders — e.g. epilepsy, mental illness, HIV infection — often prescribe a series of treatments in order to maximize favorable outcome for the patient. This generally requires modifying the duration, dose or type of treatment over time. Selecting the best sequence of treatments for an individual presents significant challenges, due to the heterogeneity in response to treatment, as well as the potential for relapse or side-effects. Clinicians often rely on clinical judgement and instinct, rather than formal evidence-based processes to optimize sequences of treatments.

Reinforcement learning (RL) is a well-known framework for optimizing sequences of actions in an evolving, time-varying system (Sutton & Barto 1998). When applied in the context of treatment design, reinforcement learning provides the means to evaluate the long-term effect of a given treatment, and thus optimize *sequences of treatments* for a given objective.

The idea of applying reinforcement learning to optimize treatment strategies is relatively novel both in the medical and machine learning communities. We attribute this in large part to a lack of appropriate sequential data (or alternately a generative model), which is a key requirement for applying reinforcement learning. This situation is rapidly changing: the medical community has a strong interest in designing studies with multiple sequential, randomized trials. (Murphy *et al.* 2006). In addition, ongoing clinical trials are evaluating the usefulness of treatment strategies that rely on automated prediction methods to trigger treatment (Kossoff *et al.* 2004), and significant attention is being devoted to developing high fidelity *in silico* models of chronic diseases (Vilar, Santana, & Uriarte 2006).

This paper examines the problem of applying reinforcement learning technology to optimize control strategies for deep-brain electrical stimulation in the treatment of epilepsy. In this case, acquiring large amounts of patient data is extremely expensive and invasive. Therefore we begin by investigating the use of batch reinforcement learning techniques to learn from *in vitro* studies of stimulation. We discuss several technical aspects of this problem, including data collection, feature extraction, function approximation, and validation from a small sample set.

It is worth pointing out that the RL methods we use throughout this paper are relatively simple and well known to the machine learning community. Nonetheless it is encouraging to see that these methods can have a meaningful impact on the optimization of treatment protocols. In particular, results of our experiments show that by using reinforcement learning, we can reduce the total amount of electrical stimulation to the brain by a factor of 10, while reducing the incidence of seizures by 25%, compared to the current best stimulation strategies in the neuroscience literature. If these results carry over to the human model of epilepsy, the impact will be substantial. Reducing the amount of stimulation means that there is less risk of damage to brain tissues, and also means that the battery onboard the neuro-stimulator has a much longer life (note that installing a new battery currently requires surgery). And of course, most important of all, reducing the incidence and duration of seizures has a significant impact on the quality of life of the patient.

Methodology for designing adaptive treatment strategies is an emerging area of interest in the medical and computational communities. The focus on multistage decision-making (rather than prediction, which has received much more attention in recent years) requires a change in perspective. Furthermore, the great deal of information available at each decision point raises several interesting challenges

for statisticians and computer scientists. An important aim of this paper is to draw the attention of the AI and machine learning community to this new area of research, and propose some interesting technical challenges that arise in this investigation.

Problem Statement

Epilepsy is the most common severe neurological disorder, affecting around 1% of the world population (Hauser & Hesdorffer 1990). Implantable electrical stimulation devices are now an important treatment option for patients who do not respond to anti-epileptic medication (Uthman *et al.* 2004). The effect has also been shown *in vitro* (i.e. in animal brain tissues) (D'Arcangelo *et al.* 2005). Very recently, researchers have started to design neuro-stimulation devices which trigger stimulation in response to an automated seizure detection algorithm (Kossoff *et al.* 2004). We propose to use reinforcement learning to directly optimize stimulation patterns of a closed-loop stimulation device, without necessarily requiring seizure prediction or detection. Figure 1 shows a schematic of our proposed approach.

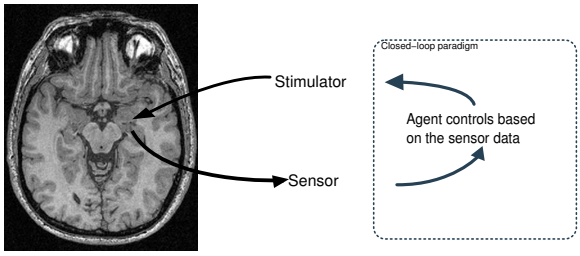

Figure 1: Reinforcement learning in deep brain stimulation.

Informally, the learning problem can be formulated as follows: at every moment in time, given some information about what happened to the signal previously (our *state*), we need to decide *which stimulation action* we should choose (if any) so as to minimize seizures *now and in the future*.

Technical Background

This section presents the technical details pertinent to our approach.

Reinforcement Learning

Reinforcement learning is a technique in which an *agent* learns to make decisions optimally in a given environment by exploring possible actions and receiving rewards for those actions. It is especially useful in situations in which the agent's environment is stochastic, and for poorly-modeled problem domains in which the optimal decision-making policy is not obvious (Kaelbling, Littman, & Moore 1996; Sutton & Barto 1998).

Formally, we model the problem as a Markov decision process (MDP) consisting of a set of states $\mathcal{S}$ and a set of actions $\mathcal{A}$ available to the agent. Time is modeled as a series of discrete steps with $0 \leq t \leq T$. On performing an action $a \in \mathcal{A}$ in state s, the agent receives a scalar reward $r = R(s,a)$ and the environment moves to a new state s' according to some conditional probability distribution $P(s'|s,a)$. The state is assumed to be a sufficient statistic for the past sensor observations. The agent's goal is to find a policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$ that maps each state to an action such as to maximize the expected total reward over some time horizon:

$$R_T = E\left[\sum_{t=0}^{T} \gamma^t r_t\right]. \quad (1)$$

Here $\gamma \in (0,1]$ is a discount factor for future rewards (it can be thought of as the agent's probability of surviving to the next time step). For $T = \infty$, γ must be less than one to preclude an infinite total reward. For finite T we can allow $\gamma = 1$.

Given this formulation, we can write the value of a given state if the agent follows a fixed policy π as:

$$V^{\pi}(s) = E_{\pi}\left[\sum_{t=0}^{T} \gamma^t r_t\right]. \quad (2)$$

We define the *optimal* value for a state $V^*(s)$ to be:

$$V^*(s) = \max_{\pi} E_{\pi}\left[\sum_{t=0}^{T} \gamma^t r_t\right], \quad (3)$$

which we can expand to the recursive equation:

$$V^*(s) = \max_{a \in \mathcal{A}} \left(R(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s,a) V^*(s') \right). \quad (4)$$

Therefore the value of a state is the maximum of the reward possible in this state plus the expected value over the successor states. The optimal policy $\pi^*(s)$ is then:

$$\pi^*(s) = \arg\max_{a \in \mathcal{A}} \left(R(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s,a) V^*(s') \right). \quad (5)$$

It is also sometimes useful to express the value of a state action pair, which we formulate as:

$$Q^*(s,a) = R(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s,a) \max_{a' \in \mathcal{A}} Q^*(s',a') \quad (6)$$

and from which the optimum value function:

$$V^*(s) = \max_{a} Q^*(s,a) \quad (7)$$

and optimum policy:

$$\pi^*(s) = \arg\max_{a} Q^*(s,a) \quad (8)$$

can be easily derived (Kaelbling, Littman, & Moore 1996).

Fitted Q Iteration Algorithm

Many traditional reinforcement learning approaches use *online* learning, in which the agent interacts with the environment dynamically and updates its policy after taking each action (Sutton & Barto 1998). However, in many medical

domains, it is not possible to train an agent entirely on-line. Normally, data will be collected in a fixed series of experimental trials and the potentially disruptive effects of an untrained agent may impose an unacceptable risk to the patient.

In cases such as this, it is preferable to utilize a *batch* mode reinforcement learning approach, in which the agent is trained using a series of previously recorded trajectories containing state, action, and reward information.

The fitted Q iteration algorithm (Ernst, Geurts, & Wehenkel 2005), which builds on earlier work on fitted value iteration (Gordon 1999; Ormoneit & Sen 2002), takes as input a set $\mathcal{F}$ of 4-tuples of the form $< s_t, a_t, r_t, s_{t+1} >$, where each tuple is an example of the one-step transition dynamics of the system. Unlike earlier formulations of batch-mode RL, the fitted Q iteration algorithm is well suited for problems with continuous state and action spaces. Also, the algorithm has been shown to make efficient use of training data (Kalyanakrishnan & Stone 2007), which is especially important in medical applications, where data may be sparse and expensive to collect.

The algorithm makes use of the recurrence relation:

$$Q_N(s_t, a_t) = R(s, a) + \gamma \max_{a' \in \mathcal{A}} Q_{N-1}(s', a'), \forall N > 1 \quad (9)$$

with $Q_1(s, a) \equiv R(s, a)$. As N increases, this sequence converges to the true Q function (Equation 6) in the infinity norm.

If we do not know the transition dynamics or reward function $R(s, a)$ of the MDP, we can still approximate Equation 9 using the fitted Q iteration algorithm. At each iteration k of the algorithm, we form an estimate $\hat{Q}_k$ of the the true Q_N function by iteratively learning the mapping:

$$\hat{Q}_k(s_t, a_t) = r_t + \gamma \max_{a' \in \mathcal{A}} \hat{Q}_{k-1}(s_{t+1}, a'). \quad (10)$$

By using the empirical return r_t in this formulation, the reinforcement learning problem can be cast as a batch supervised learning problem. Thus any regression algorithm can be used to learn the mapping $Q : S \times A \rightarrow \Re$.

Extremely Randomized Trees

The fitted Q iteration algorithm requires an appropriate supervised regression algorithm to learn the $\hat{Q}_N$ functions. In this paper we follow the example of Ernst *et al.* (2006), another medical application, and use Extremely Randomized (Extra) trees (Geurts, Ernst, & Wehenkel 2006).

Unlike classical regression tree algorithms such as CART or Kd-trees, the Extra Tree algorithm builds an ensemble of trees, and the overall value returned by the final classifier is the mean of the values of the individual trees. The algorithm has three parameters: M, the number of trees to create; K, the number of candidate tests at each node; and n_{min}, the minimal number of nodes at each leaf.

The algorithm builds each of M trees using the entire training set $\mathcal{F}$. Each node is constructed by creating K candidate tests consisting of a randomly selected element of the feature vector and a random cut point. A score is calculated for each candidate test based on the relative variance reduction of each test. The best test is kept and all others discarded. The process continues until each leaf node contains no more than n_{min} elements.

Any regression tree algorithm could be an appropriate choice of supervised regression algorithm, given both their efficiency and their excellent performance in the presence of noisy or irrelevant features. In empirical experiments with several reinforcement learning domains, the Extra Trees algorithm has exhibited excellent performance in terms of both computational efficiency and empirical return relative to other regression tree algorithms (Ernst, Geurts, & Wehenkel 2005), therefore we select this method.

Problem Definition

This section describes how the problem of controlling epileptic seizures can be formulated in the reinforcement learning framework, and in particular how we propose to apply the Extremely Randomized Trees method to optimize the choice of stimulation strategies.

Data Collection

The data used in this study are field potential recordings of seizure-like activity recorded in slices of rat brains. The slices were maintained in a bath of artificial cerebrospinal fluid containing the convulsant drug 4-aminopyridine to induce seizure-like activity. This is a standard *in vitro* model of epilepsy. The five series of recordings were made using microelectrodes inserted in the regions of interest and sampled at a rate of 2008 Hz. For our analysis we use the recordings made in the perirhinal cortex.

Electrical stimulation of the amygdala was performed on the slices in a fixed series of at least seven phases. Each series begins with a period of recording with no stimulation. Then, stimulation was applied for several minutes at 0.2 Hz. The slice was then allowed to return to baseline for a period of several minutes. This process was repeated with stimulation at 0.5 Hz, and 1.0 Hz.

Figure 2 shows a sample trace, taken during stimulation at 0.5Hz. A typical seizure appears in the first half of the trace. The stimulation actions are also visible in this recording. However, the actions may or may not be present depending on the placement of recording and stimulating electrodes.

Signal Processing Each trace was divided into a set of overlapping frames of 32768 samples (approximately 16 seconds) in length, with each frame beginning 4096 samples after the previous frame. Each frame is smoothed with a Hann window and normalized, and the mean, range, and energy of the signal is calculated. A discrete fast Fourier transform is used to extract spectral magnitude features from the frame. Within each frame, the smoothing, normalization, and Fourier transform is repeated for the final half frame, quarter frame, eighth frame, and sixteenth frame . Low frequency components are extracted from the full-frame spectrum, and high frequency components from the subframe spectra. These features are combined with the mean, range, and energy of each subframe to yield a 114-dimensional continuous feature vector.

Labeling To allow extraction of reward and action information, each trace was labeled by hand to indicate whether each frame reflected seizure or normal activity, and to record

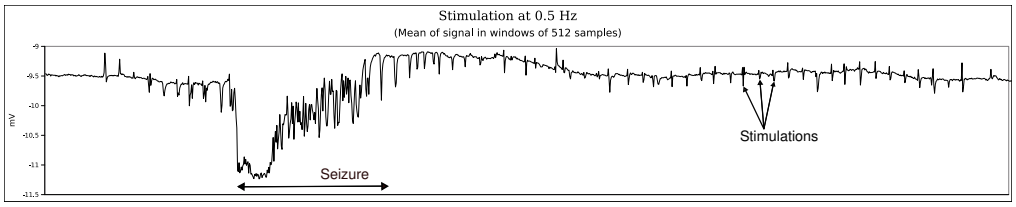

Figure 2: Trace example from the dataset

which stimulation protocol was in use at each time on each trace. Artifacts were also noted, so that they could be removed from the analysis.

Reinforcement Learning

Our state space $\mathcal{S}$ is constructed such that each element s_t is a vector of 114 continuous dimensions, summarizing past EEG activity. Our action set $\mathcal{A}$ consists of 4 options: no stimulation, and stimulation at one of the fixed frequencies of 0.2, 0.5, or 1.0 Hz. Each frame is assigned an action a_t based on the labeling information.

We define a reward function $r_t = R(s, a)$ to penalize both stimulation and seizure frames as follows:

$$r_t = \begin{cases} -1.0 & \text{if seizure and stimulation off} \\ -1.04 & \text{if seizure and stimulation on} \\ +0.01 & \text{if no seizure and stimulation off} \\ -0.03 & \text{if no seizure and stimulation on.} \end{cases} \quad (11)$$

This reward function reflects an unresolved trade-off between the cost of a seizure and the cost of stimulation. We arbitrarily chose to make the seizure events 25 times more costly than stimulation events.

Each element of the training set $\mathcal{F}$ is then constructed by concatenating the experience-tuples $< s_t, a_t, r_t, s_{t+1} >$.

For all of our experiments, the discount factor $\gamma = 0.95$.

We assume a discrete time step of 2 seconds. This is sufficient to compute our input features in real time, yet is sufficiently short to allow flexibility in the learned policy.

Training the regression trees

The procedure we use to train the trees is analogous to that proposed by Ernst *et al.* (2006). A few of the implementation details are worth mentioning.

Note that in the experiments below, we grow a set of $M = 48$ trees *for each action*. The estimate $\hat{Q}(s, a)$ is obtained by averaging the value returned by each tree in the a-th set, for the current state s.

The parameter K, the number of candidate tests created when expanding a node, was set to 30. The value of n_{min}, the minimum number of elements at each leaf, was set to 5.

Performance of the algorithm was quite robust to these parameter choices, within an order of magnitude. This is consistent with the original empirical analysis of tree-based RL (Ernst *et al.* 2006).

During the training phase, value iteration is applied over the set of trees. For the first 30 iterations, we allow the set of trees to be rebuilt entirely at each iteration. After this first phase, the structure of the trees is fixed and iterations are applied until the Bellman error falls below a given threshold. When the tree structure is fixed, only the leaf values in the trees are updated. It is necessary to fix the tree structure at some point to ensure proper convergence. Fixing the tree structure from the beginning is not desirable, as the early structure may be inadequate to reflect the final Q-function.

Note that the extremely randomized trees can be built completely in parallel since they are independent of each other. Our implementation was multithreaded to take advantage of this and allow faster learning.

Testing tree-based RL strategies

To validate our method for optimizing adaptive stimulation strategies, the obvious option is to test it directly *in vitro* on epileptic brain slices, against other strategies of stimulation. However, this approach is extremely labour-intensive, and therefore not practical as a first test of feasibility.

An easier alternative would be to use an *in silico* model of epilepsy, as is usually done to validate RL algorithms. However to date there are no good generative models of temporal-lobe epilepsy. Existing state-of-the-art models, such as that of Netoff *et al.* (2004) do not include spontaneous transition into, and out of, seizures. Furthermore they do not include any mechanisms for applying electrical stimulation. So while they are interesting from a physiological perspective, they are not useful to evaluate the effectiveness of seizure-control strategies.

So instead, we rely on some simple empirical indicators which we can calculate using a hold-out testing set, which is separate from our training data. Our original data set includes recordings from 5 animal slices. Therefore during testing we perform a 5-fold cross-validation, whereby we train on data from 4 different animal slices, and test on the 5th. This means that data in the test set comes from a different animal than the training data. It is well-documented that epileptic seizures vary greatly between animals (as well as individuals), therefore this is an important test for the generalizability of our approach.

However there is a well-known difficulty in using a test set to validate a target policy π. That is the fact that the test set was collected *under a given policy*, thus the target policy (which we wish to evaluate) cannot be *applied* on this test set. The most common solution is to use a form of rejection sampling to select only those segments of the test set which are consistent with the target policy. Recall that the test set is divided into single-step episodes: $< s_i, a_i, r_i, s_{i+1} >$. We

then use the following indicator function:

$$I_\pi(s_i, a_i) = \begin{cases} 1 \text{ if } \pi(s_i) = a_i \\ 0 \text{ otherwise} \end{cases} \quad (12)$$

to indicate that the action that would be selected by the target policy (namely $\pi(s_i)$) matches the action in the test set (namely a_i). We exclude all experience-tuples that do not match the target policy.

Empirical evaluation

Throughout our empirical validation, we consider four different scores to quantify the performance of various stimulation strategies.

The first score is an estimated proportion of seizure states when following a particular strategy π. Again, we compare the action selected by the policy and the action in the test trace for each state-action-reward tuple from the test trace, and count the number of states which were labeled as "seizure":

$$\hat{S}_\pi = \frac{\sum_{i=0}^{N} I_\pi(s_i, a_i) I_{seizure}(s_i)}{\sum_{i=0}^{N} I_\pi(s_i, a_i)}, \quad (13)$$

where $I_{seizure}(s_i)$ indicates whether state s_i was labeled as a seizure (1 if yes, 0 if no).

The second score we consider is the number of actual electrical stimulation events that would be used by a particular strategy, over the test trace. These first two scores are included because they are more common in the epilepsy literature, and thus useful metrics to gauge the potential acceptability of our method.

The third score calculates the expected immediate rewards. Formally,

$$\hat{R}_\pi = \frac{\sum_{i=0}^{N} I_\pi(s_i, a_i) r(s_i)}{\sum_{i=0}^{N} I_\pi(s_i, a_i)}, \quad (14)$$

where $s_i \in T_{test} \forall i$, $I_\pi(s_i, a_i)$ is the indicator function defined above, and $r(s_i)$ is the immediate reward associated with s_i from the labeling of the data.

The fourth score calculates the expected return (i.e. discounted sum of rewards). Formally,

$$\hat{Q}_\pi = \frac{\sum_{i=0}^{N} I_\pi(s_i, a_i) \left[r(s_i) + \gamma \hat{Q}(s_{i+1}, \pi(s_{i+1})) \right]}{\sum_{i=0}^{N} I_\pi(s_i, a_i)}, \quad (15)$$

where $\hat{Q}$ is the estimated expected value of applying a policy π. In the case of the tree-based RL method, $\hat{Q}$ is defined as in Equation 10. For fixed stimulation strategies, which were in fact deployed during data collection, we use the empirical return instead.

These last two scores are included because they reflect the actual reward function, and are more commonly used in the RL literature to validate methods. Since our reward function is a linear combination of the amount of both stimulation and seizure, these are in some sense aggregates of the other two scores.

Comparison to fixed stimulation strategies

In this section, we evaluate the performance of our tree-based reinforcement learning stimulation policy (denoted *TBRL*), in comparison to state-of-the-art stimulation strategies in the epilepsy literature. Indeed, studies of electrical stimulations have thus far focused strictly on fixed (periodic) stimulation strategies. For the particular animal model we are considering, extensive experiments have been conducted using stimulation events at 0.2Hz, 0.5Hz, and 1.0Hz, as well as observing what happens when no stimulation is applied (denoted *Control*). According to these experiments, the best fixed stimulation frequency for this type of epilepsy is 1Hz (D'Arcangelo *et al.* 2005).

Figure 3 compares the proportion of states in which a seizure is observed (according to our labels) under each of the policies. This corresponds to the score in Equation 13. We observe that TBRL is most efficient at reducing the incidence of seizures. It shows a roughly 25% improvement over the best standard stimulation policy currently used by neuro-scientists (i.e. 1Hz), and is about a 60% improvement compared to having no stimulation whatsoever (i.e. Control). This is extremely encouraging, especially given the fact that the adaptive strategy (TBRL) was trained on other animal data, and not on the test slice. We assume performance would be greatly enhanced by continuing training with the target animal. Informal results seem to confirm this (e.g. including all five slices in the training set).

Figure 4 compares the number of stimulation actions applied under each strategy. As expected, this number increases with frequency (in the case of the fixed stimulation policies). The learned policy (TBRL) uses roughly a tenth of the stimulation applied by the 1.0Hz policy. This is extremely important for two reasons: it reduces the potential for tissue damage and it significantly increases battery life of the neuro-stimulator.

Figure 5 shows the empirical reward for the various stimulation policies. Figure 6 shows the empirical return for each of the policies considered. The results here are not surprising, since in a way they represent an aggregate measure over the two scores presented in the previous figures. These results confirm that TBRL is able to simultaneously minimize the amount of both stimulation and seizure much more effectively than fixed strategies.

Comparison to other Adaptive Strategies

In this section, we explore other related methods of optimizing stimulation strategies. While our choice of the tree-based RL (TBRL) method was strongly motivated by their solid empirical evaluation in previous tasks (Ernst *et al.*

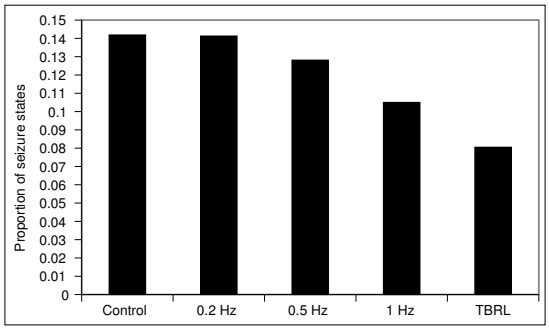

Figure 3: Proportion of seizures states (compared to non-seizure), comparing the tree-based RL method (TRBL), no stimulation (Control), and fixed stimulation strategies (0.2Hz, 0.5Hz, 1Hz).

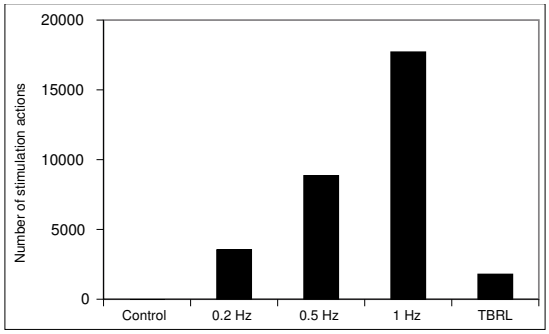

Figure 4: Total number of stimulation actions, comparing the tree-based RL method (TRBL), no stimulation (Control), and fixed stimulation strategies (0.2Hz, 0.5Hz, 1Hz).

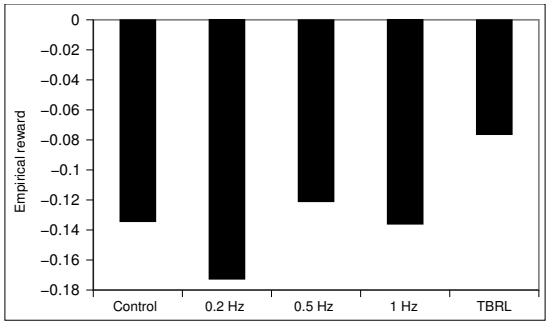

Figure 5: Empirical reward, comparing the tree-based RL method (TRBL), no stimulation (Control), and fixed stimulation strategies (0.2Hz, 0.5Hz, 1Hz).

2006), it is worthwhile verifying whether the choice is indeed appropriate for the current problem domain.

For this reason, we trained a simple neural net function approximator (denoted *NNRL*), as an alternative to the Extra Trees. The input to the neural net consists of the 114-dimensional feature vector, as described above. The output is the target Q-function, therefore we train a separate neural net for each action. Each network includes 1 hidden layer

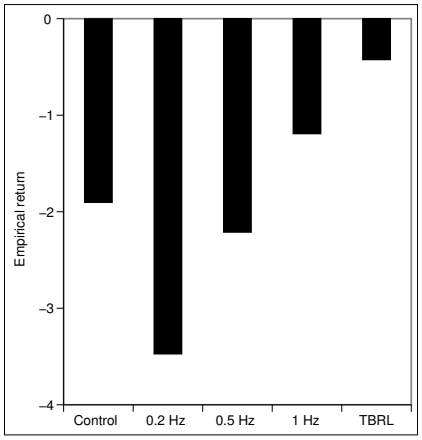

Figure 6: Average empirical return, comparing the tree-based RL method (TRBL), no stimulation (Control), and fixed stimulation strategies (0.2Hz, 0.5Hz, 1Hz).

of 80 nodes, and a single output corresponding to the target Q-function. We assume the same data and training/testing protocol as for TBRL (as described above). The main motivation for including this result is to examine our choice of function approximator; neural nets have been used extensively in the RL literature to solve challenging real-world tasks (Tesauro 1995).

In addition, we also train a second instance of TBRL, denoted *TBRL-sf*, which uses a reduced feature set, compared to the original. Rather than using all 114 input features (acquired from a full 16-second window), we consider only frequency components extracted from a shorter 2-second window, and in the 7Hz-26Hz range. In total, we preserve 40 of the 114 original features. The idea here is to investigate the effect of having a rich feature set. If we can get good results with a reduced feature set, then we can hope to improve computational speed, and possibly response rates.

Note that all results for TBRL presented in this section are identical to those in the previous section.

Figure 7 shows the proportion of time during which seizures occur, under each of the methods listed above. We see once again that the tree-based method is most efficient at controlling seizures. The method is also quite robust to a drastic reduction of its feature set. This confirms earlier results showing that the extremely randomized trees are robust to the presence of extraneous features. The neural net learner performs quite poorly (on par with the *Control* policy of Figure 3). We tried various configurations for the neural net (# of hidden units, learning rate), without any improvement. In some cases, results were much worse.

Figure 8 shows the amount of stimulation applied by each method. We see that the neural net appears to vastly overstimulate. We do not see any obvious reasons for this, and presume the learner has reached a local minimum.

Finally, Figure 9 shows the empirical reward for each of the adaptive stimulation policies. [1] As expected (based on

[1] We do not show the empirical return in this case. The estimate

the two previous figures), the tree-based methods have the best scores, whereas the neural net performs quite poorly.

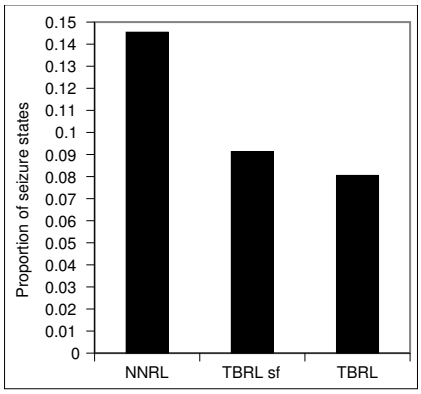

Figure 7: Proportion of seizures states (compared to non-seizure), comparing the tree-based RL method (TRBL), tree-based RL with a reduced state set (TBRL sf), and a neural net learner (NNRL).

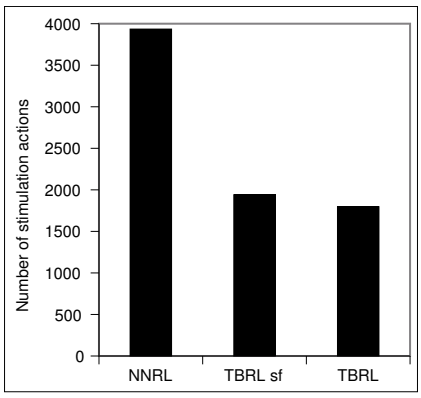

Figure 8: Total number of stimulation actions, comparing the tree-based RL method (TRBL), tree-based RL with a reduced state set (TBRL sf), and a neural net learner (NNRL).

We conclude our empirical evaluation by looking at some sample traces, illustrating the TBRL policy in action. Figure 10 shows a sample trace from the test set, along with the TBRL strategy chosen. In this segment, no actions were applied during the actual data collection. Recall that TBRL must choose between 4 actions: no simulation, 0.2 Hz stimulation, 0.5 Hz stimulation, and 1.0 Hz stimulation. In effect, the TBRL policy is just a mixture of these fixed policies. We see in Figure 10 that the amount of stimulation increases significantly during a seizure, and continues intermittently afterwards. Figure 11 shows similar results for another excerpt from the dataset. In this case, there is significantly more pre-seizure activity (characterized by the short

of the return for the NNRL method includes a large bias term (due to the fact that the Q-function did not converge to the correct value), so the return estimates are not directly comparable between the two TBRL policies, and the NNRL policy.

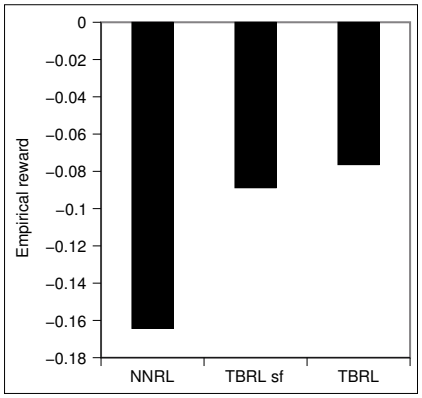

Figure 9: Empirical reward, comparing the tree-based RL method (TRBL), tree-based RL with a reduced state set (TBRL sf), and a neural net learner (NNRL).

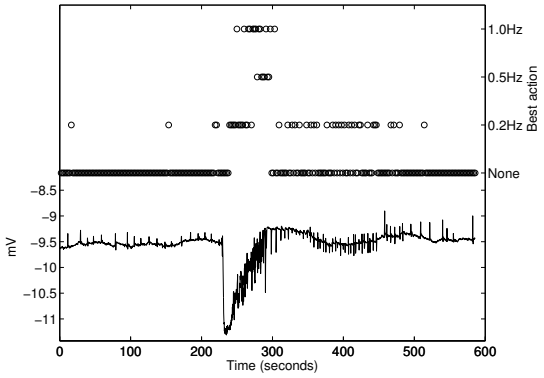

Figure 10: Sample data trace and TBRL policy #1

spikes seen before and after the seizure), and consequently, the adaptive policy responds by increasing the amount of stimulation throughout (both before and after the seizure).

Discussion

This paper provides encouraging evidence that reinforcement learning may be an important technique for optimizing sequential treatment strategies for chronic diseases, and in particular for epilepsy. The results obtained so far show that an adaptive stimulation strategy, trained from batch data, substantially outperforms fixed stimulation policies, which have been the norm in the epilepsy literature. In particular, results of our experiments show that by using reinforcement learning, we are able to reduce the incidence of seizures by 25%, compared to the current best stimulation strategies in the neuroscience literature (and 60% compared to when there is no stimulation). Furthermore, the total amount of electrical stimulation to the brain is reduced by a factor of about 10. If these results carry over to the human model of epilepsy, the impact for patients will be substantial. The anticipated benefits from reducing the incidence of seizures are well-known; we note that the reduction in stimulation would also have a direct impact on the quality of life of epilepsy

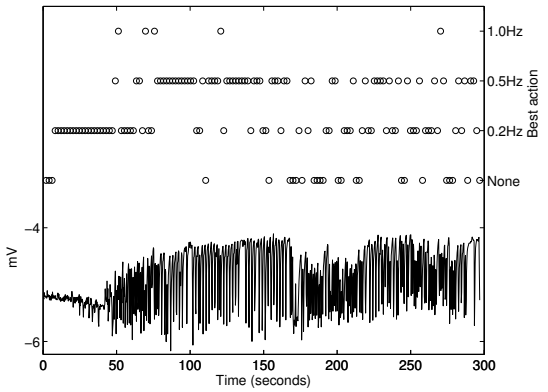

Figure 11: Sample data trace and TBRL policy #2

patients since electrical stimulation accounts for approximately 50% of the power consumption in neuro-stimulators, and batteries can only be changed through surgery.

The results presented above were obtained using data from an *in vitro* model of epilepsy. We are now planning a series of experiments, whereby the adaptive stimulation strategy learned using the batch data will be evaluated online, using live *in vitro* slices. Performing such experiments is very time-consuming and expensive. This highlights the value of developing good computational models of dynamical diseases. Such models exist for some diseases, such HIV/AIDS and cancer, however none are currently available for epilepsy. This may be due to the highly unpredictable nature of the disease. This presents interesting challenges related to statistical modeling and inference.

Most of the reinforcement learning methodology leveraged in this paper is well known in the AI community. In particular, fitted Q-iteration with Extra Trees has been extensively tested in standard RL simulation tasks (Ernst, Geurts, & Wehenkel 2005), as well as clinical tasks (Ernst *et al.* 2006). The results presented here confirm the approach performs very well, even with complex, multi-dimensional input. Furthermore it seems robust to extraneous variables and other sources of noises, much more so than neural nets which performed quite poorly in the problem domain.

In conclusion, this paper presents a novel application of reinforcement learning methodologies to a challenging and important optimization problem. The potential impact of this work is tremendous, and while the early results are promising, there remains a long road of empirical validation. Along the way, many interesting computational questions will arise, including: *How should we quantify performance of adaptive strategies*? *How we can learn from very little training data*? *Can we design "safe" exploration policies, with formal guarantees on worse-case performance*? *How can we re-use data, or learned policies, between different patients*? These are just a few of the questions which are pertinent to the task at hand. Many of these have been on the agenda of AI researchers for a number of years. We hope this paper encourages them to continue investigating these challenging issues, as well as look towards applications pertaining to adaptive treatment strategies to motivate and validate their research endeavours.

Acknowledgments The authors gratefully acknowledge financial support from NSERC and CIHR.

References

D'Arcangelo, G.; Panuccio, G.; Tancredi, V.; and Avoli, M. 2005. Repetitive low-frequency stimulation reduces epileptiform synchronization in limbic neuronal networks. *Neurobiology of Disease* 19(1-2):119–128.

Ernst, D.; Stan, G.-B.; Gonçalves, J.; and Wehenkel, L. 2006. Clinical data based optimal STI strategies for HIV: A reinforcement learning approach. In *15th Machine Learning Conference of Belgium and The Netherlands*.

Ernst, D.; Geurts, P.; and Wehenkel, L. 2005. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research* 6:503–556.

Geurts, P.; Ernst, D.; and Wehenkel, L. 2006. Extremely randomized trees. *Machine Learning* 63(1):3–42.

Gordon, G. J. 1999. *Approximate Solutions to Markov Decision Processes*. Ph.D. Dissertation, Carnegie Mellon University, Pittsburgh, PA.

Hauser, W. A., and Hesdorffer, D. C. 1990. *Epilepsy: Frequency, Causes and Consequences*. New York: Demos.

Kaelbling, L. P.; Littman, M. L.; and Moore, A. W. 1996. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research* 4:237–285.

Kalyanakrishnan, S., and Stone, P. 2007. Batch reinforcement learning in a complex domain. In *The Autonomous Agents and Multiagent Systems Conference*.

Kossoff, E. H.; Ritzl, E. A.; Politsky, J. M.; Murro, A. M.; Smith, J. R.; Duckrow, R. B.; Spencer, D. D.; and Bergey, G. K. 2004. Effect of an external responsive neurostimulator on seizures and electrographic discharges during subdural electrode monitoring. *Epilepsia* 45(12):1560–1567.

Murphy, S. A.; Oslin, D.; Rush, A.; and Zhu, J. 2006. Methodological challenges in constructing effective treatment sequences for chronic psychiatric disorders. *Neuropsychopharmacology* 32(2):257–262.

Netoff, T. I.; Clewley, R.; Arno, S.; Keck, T.; and White, J. A. 2004. Epilepsy in small world networks. *Journal of Neuroscience* 24(37):8075–8083.

Ormoneit, D., and Sen, S. 2002. Kernel-based reinforcement learning. *Machine Learning* 49:161–178.

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press.

Tesauro, G. 1995. Temporal difference learning and td-gammon. *Communications of the ACM* 38(3):58–68.

Uthman, B. M.; Reichl, A. M.; Dean, J. C.; Eisenschenk, S.; Gilmore, R.; Reid, S. A.; Roper, S. N.; and Wilder, B. J. 2004. Effectiveness of vagus nerve stimulation in epilepsy patients: A 12 year observation. *Neurology* 63:1124–1126.

Vilar, S.; Santana, L.; and Uriarte, E. 2006. Probabilistic neural network model for the in silico evaluation of anti-hiv activity and mechanism of action. *Journal of Medicinal Chemistry* 49(3):1118–1124.