Online Learning for the Control of Human Standing via Spinal Cord Stimulation

Thesis by
Yanan Sui

In Partial Fulfillment of the Requirements for the
degree of
Doctor of Philosophy

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2017
Defended December 15, 2016

ACKNOWLEDGEMENTS

I’m very grateful to my advisor Professor, Joel Burdick, for the support throughout my PhD study. Joel introduced me to a variety of areas, provided the vision, and gave me the freedom and support to pursue my projects.

I would like to sincerely thank Professor Yisong Yue and Professor Andreas Krause for being great mentors for me. The thesis would have not been possible if it were not for their guidance and collaborations.

Special thanks to Prof. Richard Murray, Prof. Pietro Perona, Prof. Yaser Abu-Mostafa, and Prof. Yu-Chong Tai, who are/were on my research committee, for their helpful discussions and suggestions.

I would like to thank all the colleagues in Joel’s robotics group and the dolcit group. I also would like to thank other professors who have helped me grow up in the past years. Furthermore, I would like to thank my friends and teachers at Caltech.

Last, I would like to give my deepest gratitude to my family for their endless love and support.

ABSTRACT

Many applications in recommender systems or experimental design need to make decisions online. Each decision leads to a stochastic reward with initially unknown distribution, while new decisions are made based on the observations of previous rewards. To maximize the total reward, one needs to balance between exploring different strategies and exploiting currently optimal strategies within a given set of strategies. This is the underlying trade-off of a number of clinical neural engineering problems, including brain-computer interface, deep brain stimulation, and spinal cord injury therapy. In these systems, complex electronic and computational systems interact with the human central nervous system. A critical issue is how to control the agents to produce results which are optimal under some measure, for example, efficiently decoding the user's intention in a brain-computer interface or performs temporal and spatial specific stimulation in deep brain stimulation. This dissertation is motivated by electrical sipnal cord stimulation with high dimensional inputs(multi-electrode arrays). The stimulation is applied to promote the function and rehabilitation of the remaining neural circuitry below the spinal cord injury, and enable complex motor behaviors such as stepping and standing. To enable the careful tuning of these stimuli for each patient, the electrode arrays which deliver these stimuli have become increasingly more sophisticated, with a corresponding increase in the number of free parameters over which the stimuli need to be optimized. Since the number of stimuli is growing exponentially with the number of electrodes, algorithmic methods of selecting stimuli is necessary, particularly when the feedback is expensive to get.

In many online learning settings, particularly those that involve human feedback, reliable feedback is often limited to pairwise preferences instead of real valued feedback. Examples include implicit or subjective feedback for information retrieval and recommender systems, such as clicks on search results, and subjective feedback on the quality of recommended care. Sometimes with real valued feedback, we require that the sampled function values exceed some prespecified "safety" threshold, a requirement that existing algorithms fail to meet. Examples include medical applications where the patients' comfort must be guaranteed; recommender systems aiming to avoid user dissatisfaction; and robotic control, where one seeks to avoid controls that cause physical harm to the platform.

This dissertation provides online learning algorithms for several specific online decision-making problems. SelfSparring optimizes the cumulative reward with relative feedback. RankComparison deals with ranking feedback. SafeOpt considers the optimization with real valued feedback and safety constraints. Correlational Dueling is designed for specific spinal cord injury therapy.

A variant of Correlational Dueling was implemented in closed-loop human experiments, controlling which epidural stimulating electrodes are used in the spinal cord of SCI patients. The results obtained are compared with concurrent stimulus tuning carried out by human experimenter. These experiments show that this algorithm is at least as effective as the human experimenter, suggesting that this algorithm can be applied to the more challenging problems of enabling and optimizing complex, sensory-dependent behaviors, such as stepping and standing in SCI patients.

In order to get reliable quantitative measurements besides comparisons, the standing behaviors of paralyzed patients under spinal cord stimulation are evaluated. The potential of quantifying the quality of bipedal standing in an automatic approach is also shown in this work.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Number *Page*

Chapter 1

INTRODUCTION

We consider the problem of sequential decision-makings under uncertainty, where we seek to optimize an initially unknown function from noisy samples. This requires balancing exploration (learning about the objective) and exploitation (localizing the maximum), a problem well-studied in the multi-armed bandit literature. This is the underlying trade-off of a number of clinical neural engineering problems, including brain-computer interface, deep brain stimulation, and spinal cord injury therapy. In these problems, (high dimensional) control systems interact with the human central nervous system. A critical issue is how to control these agents to generate optimal controlling inputs. This dissertation is concerned with electrical spinal cord stimulation via multi-channel electrode arrays. The stimulation promotes the function and rehabilitation of the remaining neural circuitry below the injury, with the goal of enabling human motor behaviors such as standing and stepping. The electrode arrays which are used to tune these stimuli for each patient have a large number of freely tuning parameters, and little prior knowledge can be applied to restrain the exponential growth of input space w.r.t. the number of electrodes. Due to the expense and relative inaccessibility of expert hand-tuning, a more strategic, algorithmic method of selecting stimuli is necessary.

In this dissertation, we study online learning algorithms for several specific online decision-making problems. SelfSparring optimizes the cumulative reward with relative feedback. RankComparison deals with ranking feedback. SafeOpt considers the optimization with real valued feedback and safety constraints. Correlational Dueling is designed for the specific spinal cord injury therapy.

A variant of Correlational Dueling was implemented in closed-loop human experiments, controlling which epidural stimulating electrodes are used in the spinal cord of SCI patients. The results obtained are compared with concurrent stimulus tuning carried out by human experimenter. These experiments show that this algorithm is at least as effective as the human experimenter, suggesting that this algorithm can be applied to the more challenging problems of enabling and optimizing complex, sensory-dependent behaviors, such as stepping and standing in SCI patients.

Studying stimulation induced standing for paralyzed patients brings many challenges that require concurrent use of advanced tools in imaging techniques, computer simulations, and robotics. Biomechanics is aiming to analyze movement to enhance performance and understand mechanisms of injury. In robotics research, similar efforts concentrate on improving the dynamic performance of humanoid robots and other redundant mechanisms without compromising safety. The effectiveness of control relies on robust feedback measures. In order to get reliable quantitative measurements besides comparisons, the standing behaviors of paralyzed patients under spinal cord stimulation are evaluated. The potential to quantify the quality of bipedal standing in an automatic approach is also shown in this thesis.

1.1 Motivation

This dissertation is motivated by spinal cord injury therapies. Spinal cord injury (SCI) is a traumatic injury to the spinal cord resulting in losses of functions. The goal of the SCI therapies is to help patients with severe spinal cord injury recover their ability to stand, walk, and regain voluntary movements.

According to the 2016 report by the National Spinal Cord Injury Statistical Center, the number of people in the U.S. who are alive in 2016 who have SCI has been estimated to be approximately 282,000 persons, with a range from 243,000 to 347,000 persons. Given the current population size of 314 million people in the U.S., the estimate showed that the annual incidence of spinal cord injury is approximately 54 cases per million population in the U.S. or approximately 17,000 new SCI cases each year. In particularly, about 30 percent of the cases are complete SCI, which means all the sensation and voluntary control of some parts of the body is lost. As shown in Figure 1.1, SCI is mainly caused by motor vehicle crashes (38%), followed by falls (30.5%) and violence (13.5%), and there is a quire large percentage of complete (severe) injuries (33.3%).

Spinal cord injury cannot be cured under current medical treatments, and rehabilitation can be very difficult. Besides the pain and extreme high cost of lifetime medical care, people with severe SCI cannot currently stand or walk, which impacts their social and work life. The work done for this thesis is part of a collaborative effort by Caltech, University of Louisville, and UCLA to provide new therapies for SCI. Some previous work along this line of research can be found in T. A. Desautels (2014) and Z. Liu (2016).

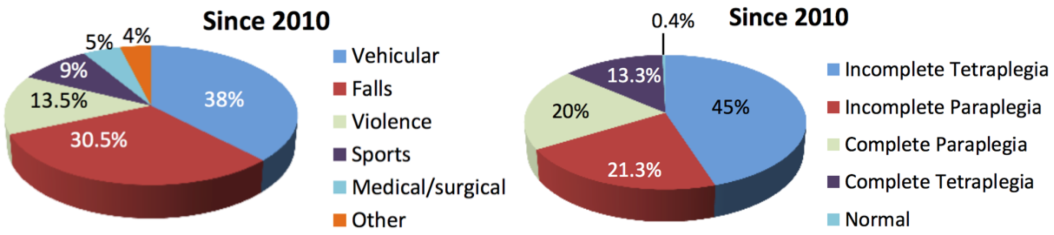


Figure 1.1: Facts on Spinal Cord Injury. The left pie chart shows the etiology of injuries. The right chart shows the neurological level and extent of lesion.

1.2 Epidural electrical stimulation

The technique we study for SCI therapy is Epidural Electrical Stimulation (EES). EES involves electrically stimulating the spinal cord via an electrode or multi-electrode array placed in the epidural space. Spinal electrical stimulation has been applied for a number of purposes, including the the alleviation of chronic pain (Shealy et al., 1967a,b). It is also used for the treatment of motor deficits, such as cerebral palsy. Recent stimulators provide more flexibility with more electrodes and more complex stimuli waveforms are applied on the electrodes. These increased capabilities allow complex stimuli to be customized after the implantation. A single device can thus accommodate changes in the stimuli as the optimal parameters change with time, as well as variations in surgical placement, injury, and patient-specific needs for symptom alleviation.

Mechanistically, SCI therapy by EES is intended to promote activity, particularly closed-loop activity, of the spinal cord below the site of the injury. This is ac-

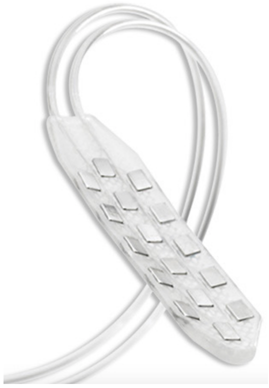


Figure 1.2: Medtronic 16-channel electrode arrays (Model 5-6-5 Specify).

complished by applying a tonic electrical stimulus to activate specific networks and structures in the spinal cord. This stimulus is typically not intended to drive the desired activity directly. Instead, stimuli enable the patient's native neural circuits to regulate motor activity according to the sensory environment of the patient, such that the muscle contractions are appropriate to the environment and behavior of the patient, such as weight shifts during standing. Such circuitry does in fact remain intact, if quiescent, below the site of the injury; an autonomously rhythm-generating structure known as a central pattern generator (CPG) is known to exist in a variety of species, including rats and cats, and is thought to exist in humans (Yury Gerasimenko, Roy, and Edgerton, 2008). These neural circuits drive and control complex motor behaviors such as stepping, even in the absence of input from the brain. EES has been applied to stimulate these networks of neurons, enabling stepping and standing after SCI (Harkema et al., 2011; Brand et al., 2012). From a control-theoretic perspective, the EES system is not intended to be the controller to the body's plant or process. The EES system modifies the activity of the intrinsic spinal controller or replaces the absent supraspinal control signal. In order to make the EES system a higher-level controller for the spinal cord and lower body system, it is necessary to measure the performance of the spinal cord and body and use these experimental measurements to make decisions about how to change the EES parameters. As the number of electrodes and free parameters increases, however, it is necessary to develop more advanced methods of selecting the stimuli delivered by EES arrays. The motivating problem of this dissertation is optimizing the stimulus patterns for the complex arrays available now and in the near future.

1.3 Learning and Optimization

To optimize the stimulation patterns, we consider efficient online learning by actively sampling the input space. Active learning techniques are algorithms for actively, rather than passively, attempting to learn about a system. An active learning algorithm is able to interactively query the user (or some other information source) to obtain the desired outputs at new data points. It is also called optimal experimental design in some statistics literature. In the traditional formulation, the active learner is interacting with an oracle, a system which accepts experimental interrogations, each single one of which is called a query and returns observations which correspond to the queries. In this fashion, the learner gradually acquires information about expected the response to any query. The element which makes this interaction active, rather than passive, is that the active learner has choices, most typically the choice of which query to submit to the oracle at each opportunity. The active learner makes these choices on the basis of a model of the oracle, constructed based upon the data. Using the choice of which queries to submit, rather than waiting passively for whatever data happens to arrive, the active learner is thus able to acquire the desired information (or simply more information) about the oracle in fewer observations than a passive learner.

Active learning can be targeted to particular pieces of information about the oracle. One important example is the problem of Bayesian optimization (BO). An interesting approach to this problem is taken by Hennig and Schuler (2012), who also make interesting points regarding the appropriate algorithm design philosophy for this setting. In this case, the learner is given a finite budget of queries and is asked to spend these queries to find the action within the available set which yields the maximum value of the reward. After the learning process is complete, the optimal action will be chosen and the algorithm will receive this reward. In order to be effective, an algorithm must observe the reward values which would be associated with the queries it has so far submitted and then choose future queries which are likely to decrease its uncertainty about where the optimum lies. Choosing queries online, rather than a priori, allows the algorithm to target these queries to the regions of the set of possible actions which appear promising on the basis of the data being acquired.

An important and closely related problem is the exploitation-exploration tradeoff. If the algorithm receives reward for each and every single query, rather than having a distinct search phase in which no rewards are obtained, followed by an exploit phase (as in the BO setting), it is important to choose these queries not simply to learn about the best rewards which may be obtained, but also to obtain high reward at this very moment. This is particularly appropriate in the SCI therapeutic setting, in which each EES stimulus and each interval of training time is valuable and should be spent intelligently. Algorithms for solving this problem have traditionally been explored under the framework of multi-armed bandits (Robbins, 1952). These algorithms make sequential decisions by trading off exploitation of actions known to yield high reward action with exploration of novel or poorly-understood actions. If these competing imperatives are properly balanced, it can sometimes be demonstrated that the algorithm will converge to the optimal action (i.e., the rate of sub-optimal actions approaches zero) with high probability in the limit of infinite time. Recent work in this field has brought bandits and Bayesian optimization together, yielding algorithms which seek to explore and exploit over very large decision sets, using models of the response function (e.g., the GP-UCB algorithm of Srinivas et al. (2010a), which uses Gaussian processes to model the reward function).

1.4 Major Problem

In order to make a practical, fully-implantable system to apply epidural electrical stimulation which is highly effective for SCI therapy, it is necessary to create, implement, and test a class of active learning algorithms with the following properties:

- Exploit the structure of the epidural spinal stimulation problem, i.e., the anatomical and neurophysiological knowledge of the spinal cord and the lower limbs, as well as the capabilities and construction of the stimulating device, to learn the responses of the patient's spinal cord and muscle activity to epidural electrical stimulation.
- Use such a model to choose queries or experimental actions in a way which enables the response functions to be learned in a query-efficient manner, due to the expense of individual queries and the large size of search space.
- Performing effective therapy for the patient, as measured by metrics of success available on a per-trial basis.
- Performing effective therapy for the patient, as measured by relative feedback such as rank or pairwise comparison.

In order for the learning process to be both efficient and tractable, the first property is necessary. To provide an effective therapy, prior information must be combined with measurements taken for this particular individual. This prior information, largely invariant, structural, and qualitative in nature, is the result of many years of neurophysiological studies and clinical experience and represents a tremendous resource for exploitation by an automated agent. Since it is desirable for this agent to accomplish the same tasks as would normally be performed by experienced clinicians, incorporating this prior information is a crucial first step. The budget of experiments, constrained principally by the time required to perform the desired measurements, but also the monetary expense of doing so, will often be several orders of magnitude smaller than the number of potential stimuli; thus, stimuli which will likely be ineffective must be rapidly eliminated from consideration, such that experimental effort is concentrated on stimuli which are more likely to be therapeutically useful. This motivates the second requirement. The third property is required by the fact that the calibration sessions in which the algorithm is run will also constitute a substantial part of the patient's therapy, and indeed, are arguably the most therapeutically useful sessions available due to the very expensive presence of highly trained clinicians and therapists. Optimally, all stimuli ever administered (including those delivered by the stimulator as the patient undertakes the tasks of daily living) should be evaluated in terms of their functional performance, such that an algorithm which takes full advantage of this opportunity for experimentation and learning may be preferable. If the algorithm operates continuously, it must treat the therapeutic effectiveness of the stimuli delivered as a substantial component of its decision-making if an effective therapy is to be applied. Furthermore, poor stimuli (those which produce low reward values, indicative of poor therapeutic performance) may produce high fatigue or confound the results of later experiments. Poor stimulus choices destroy much of the utility of the experimental or therapeutic training session. The fourth property allows for much greater flexibility in applications; the requirement of algorithms like GP-UCB that all observations be available before the next action can be selected, and thus that only one action can be pending at any time can prove to be a substantial encumbrance. In the SCI therapy setting, the data processed into the performance metric used by the algorithm is often complex and time-consuming to calculate, resulting in substantial delays between the performance of an experiment and the availability of the assessed performance on that experiment. Motion capture, for example, may take extensive hand annotation to analyze fully, and multi-channel EMG may take several minutes to process into a useful form. However, it is most efficient to assemble an experimental session which consists of an unbroken sequence of requested stimuli; this necessitates either a batch procedure or delayed selection of stimuli. Further, it is highly desirable that an active learning system have the following additional properties:

- It has rigorous guarantees of behavior, at least under some conditions.
- It is sufficiently modular to enable adaptation to different experimental conditions, e.g. by the revision of structural assumptions, the inclusion or exclusion of stimuli within the decision set, and possibly even modification of the decision rule.
- The predictions made by and the assumptions encoded within the algorithm are human interpretable.

- The computations compose the modeling and action selection steps of the algorithm should be as efficient as possible, with an eye toward deployment on systems with limited computational power, i.e., miniaturized fully implantable devices.

These secondary specifications also describe important capabilities. Guarantees of performance are an important requirement, as the algorithm's practical performance may be easier to understand in light of these guarantees. Modularity is highly desirable because various components can be interchanged to suit the particular problem at hand. From a practical perspective, modularity is also useful because it enables the re-use of computer code between similar experiments, as well as potentially allowing rigorous comparisons of different modules, e.g., model selection on the Gaussian process kernel functions. The third desire, human interpretable predictions, is important for both contributing to the body of clinical and neuroscience literature on the spinal cord, as well as verification of these models by clinical observation and experience. Finally, computational efficiency is important, as the long term goal of a fully implantable, autonomous device which administers a dynamic, data-driven therapy requires algorithms which can be run with extremely limited computational resources.

1.5 Contributions

This dissertation develops both online learning algorithms for several specific online decision-making problems and clinical experiments for the spinal cord injury therapy via electrical stimulation.

SelfSparring optimizes the cumulative reward with relative feedback. RankComparison deals with ranking feedback. SafeOpt considers the optimization with real valued feedback and safety constraints. Correlational Dueling is designed for the specific spinal cord injury therapy.

A variant of Correlational Dueling was implemented in closed-loop human experiments, and used to control which epidural stimulating electrodes are used in the spinal cord of SCI patients. The results obtained are compared with concurrent stimulus tuning carried out by human experimenter. These experiments show that this algorithm is at least as effective as the human experimenter, suggesting that this algorithm can be applied to the more challenging problems of enabling and optimizing complex, sensory-dependent behaviors, such as stepping and standing in SCI patients.

In order to get reliable quantitative measurements besides comparisons, the standing behaviors of paralyzed patients under spinal cord stimulation are evaluated. The potential to quantify the quality of bipedal standing in an automatic approach is also shown in this work.

1.6 Organization

Some background materials relevant to this dissertation, including spinal cord injury therapy, bandit problem, Gaussian processes, kernel functions, and the learning algorithms, are introduced in Chapter 2. The theoretical properties of new proposing dueling bandit algorithms are examined in Chapter 3. These results are examined by a series of computational experiments comparing these algorithms with several others. Chapter 4 presents the development of RankComparison algorithm on the dueling bandit framework and the primary experiments in complete SCI patients. The safety constraint of stimulation is studied in Chapter 5 with real valued feedback. Chapter 6 describes human experiments with correlational dueling feedback. Measuring the quality of bipedal standing in an automatic approach is explored in Chapter 7. Chapter 8 makes some final conclusions with regard to the present work and also discusses potential extensions of current results.

Chapter 2

BACKGROUND

This chapter reviews the background knowledge which is useful for the following chapters.

Section 2.1 introduces key clinical problems associated with spinal cord injury. Section 2.2 describes the current therapeutic approaches for spinal cord injury. Some background on active learning and the bandit problem are provided in Section 2.3, which is the theoretical foundation of the algorithmic developments in this thesis. Section 2.4 describes the dueling bandit problem, which uses relative feedback for our optimization. In Section 2.5, the Gaussian processes (GPs) are reviewed. The following Section 2.6 provides background on the covariance functions.

2.1 Spinal Cord Injury

Patients with SCI present different clinical symptoms depending on the location of the injury within the spinal cord, including a variety of syndromes which are symptomatic of damage to different structures within the spinal cord. Sufficiently severe damage to the spinal cord can result in the loss of voluntary control (frequently accompanied by loss of sensation as well) of the legs (paraplegia) or the legs and arms (quadraplegia). The severity of a patient's injury is most commonly assessed on the ASIA (American Spinal Injury Association) scale, as well as by the neurological level of the injury in the spinal cord, diagnosed via the affected dermatomes and myotomes, which correspond in a fixed fashion to spinal levels.

The rahabilitation after spinal cord injury has been studied in both human patients (e.g., Harkema et al. (2011)) and animal models (e.g., Brand et al. (2012)). There are several good reviews on this topic such as Thuret, Moon, and Gage (2006), Edgerton et al. (2006), and Yury Gerasimenko, Roy, and Edgerton (2008).

The primary, long-term result of this damage is a substantial loss of function in terms of motor control and sensation, resulting in impaired mobility and independence. While the symptoms of some patients improve over the first $1 \sim 1.5$ years after injury, these improvements eventually cease (Fawcett et al., 2007). The remaining deficits in sensory and motor function are at present generally considered to be largely irreversible, i.e., there is no cure for SCI, though a number of approaches have been developed which have produced gains for some patients. Interestingly, in incomplete injuries and within the general realm of motor control, the degree of recovery in the performance of individual motor tasks may be somewhat independent; this may be due to different levels of supraspinal control exercised in different motor functions, e.g., locomotion versus reaching (Grégoire Courtine et al., 2005).

Beyond loss of motor control and bodily sensation, a number of other problems commonly arise for SCI patients, particularly issues resulting from lack of exercise and from the disruption of the nervous system's internal communications. These deficits can include muscle atrophy and spasticity, as well as potentially life-threatening autonomic problems such as failures of temperature regulation and autonomic dysreflexia. For a discussion of the many and varied autonomic deficits which result from SCI, the symposium proceedings edited by Weaver and Polosa (2006) are an excellent resource.

Advances in care have meant that SCI patients who do not die immediately tend to survive for a many years, such that therapies which partially alleviate some of their symptoms are highly desirable. A survey of 681 SCI patients was reported by Anderson (2004). They found that among the seven options presented on the survey instrument, a near-majority of quadriplegics believed that recovery of hand and arm function would produce the greatest improvement in their quality of life, while a plurality of paraplegics believed that recovery of sexual function would most greatly improve the quality of theirs. A very substantial number of both populations ranked the item composed of bladder, bowel, and autonomic dysreflexia as one of their top two potential greatest gains in quality of life. Among paraplegics, walking movement, described by the survey's creator as inclusive of standing and other forms of exercise, also ranked highly, but its share of first or second votes was much lower among quadraplegics. Even given the limitations of the survey, it is striking that bladder, bowel, and autonomic dysreflexia concerns were so important, particularly as compared with walking and mobility.

Current and experimental therapies have begun to deliver this desirable alleviation of secondary symptoms; for example, the initial patient in an epidural electrical stimulation study (Harkema et al., 2011), whose therapy program included epidural electrical stimulation, locomotion training, and stand training, reports that his gains have included improved mental well-being, improved bladder and bowel function, some improvement in sensory function, some improvement in sexual function, substantial gains in muscle mass, including gains in the legs, core, and upper body, and better postural control. Note that this patient had already participated in an intensive locomotion training program, and that these gains are relative to his condition after that program. Additionally, this patient has recovered some gross voluntary motor control of his lower limbs; it has been suggested that this control is a result of use-dependent plasticity of spared supraspinal axonal projections.

2.2 Existing Therapeutic Approaches

Since no cure currently exists for spinal cord injuries, current practice focuses on therapy, applied in a variety of approaches. Some techniques attempt to directly create the pattern of muscle activation in the extremities which would ordinarily be associated with the desired activity. Other approaches focus on rehabilitating the spinal cord; Bradbury and McMahon (2006) describe these as attempting either to induce regeneration of damaged axons, repairing the damage to some extent, or to rehabilitate the spinal cord's ability to control the body without addressing the injury itself. A review of a number of approaches is presented in the following sub-sections, including epidural electrical stimulation, the approach which is the particular focus of this dissertation.

Functional Electrical Stimulation

Functional Electrical Stimulation (FES, Liberson et al. (1961)) attempts to treat the symptoms of paralysis via direct stimulation of the muscles themselves; in FES, electrical stimulators are placed on or within the skeletal muscles and are then activated in a pattern engineered to replicate a desired activity, e.g., the stride cycle. The pattern of muscle activation is directly controlled, such that FES can be used to treat foot drop in hemiplegic patients (Liberson et al., 1961), and produce weight-bearing locomotion in paraplegics (Klose et al., 1997). Applied as an exercise therapy, FES has been shown to confer gains in a variety of cardiorespiratory and metabolic metrics (Davis et al., 2008). However, since FES is an open-loop control method, the stimulation pattern must be carefully designed and/or user-controlled if complex behaviors are desired (e.g., in hand control, as examined by Mangold et al., 2004). Further, the resulting muscle contractions do not respond directly to sensory feedback, an important consideration when considering activities which require significant feedback control, e.g., standing. Another important problem with FES is rapid fatigue; muscle contraction force typically decreases rapidly under FES (see Thrasher et al., 2005, for a discussion of fatigue and the difficulties in mitigating it).

Regenerative Therapies

Another major approach to SCI rehabilitation aims to induce the spinal cord to repair itself via the introduction of signaling molecules and/or the suppression of endogenous signaling (Karimi-Abdolrezaee et al., 2012), or to introduce cells, exogenous or autologous, into the injury site which would promote or support regrowth (Coumans et al., 2001; Wu et al., 2012). Often, scaffolds are constructed from biomaterials and used to support regrowth by providing a stable and permissive environment (Pego et al., 2012). Regenerative therapies seem promising in the long-term, but have not to date met with substantial success in patients with complete SCI (Thuret, Moon, and Gage, 2006). However, there is evidence that, in the case of some incomplete injuries, and even without therapeutic aid beyond exercise, the central nervous system can reroute connections through existing neurons which bypass the injury site to make some small functional gains (Bareyre et al., 2004; Gregoire Courtine et al., 2008). If these gains can be further improved, they may provide the basis for substantial recovery in the future.

Cord-Rehabilitative Approaches

In contrast to FES and regenerative therapies, cord-rehabilitative approaches do not seek to directly drive the muscles or repair the spinal cord; rather, these approaches take a middle road and attempt to modify the function of the spinal cord in order to produce the desired motor behavior. This avenue of SCI therapy attempts to take advantage of the surviving spinal cord circuitry below the site of injury, which remains viable and adaptable (Edgerton et al., 2006). Specific targets include interneuron networks responsible for reflexes and the central pattern generator, the region of the spinal cord responsible for generating the overall pattern of muscle activation in walking (Dimitrijevic, Yuri Gerasimenko, and Pinter, 1998).

Methods of this type may attempt to use any of a variety of approaches to promote lower spinal cord activity. Pharmaceutical replacement of or substitution for neurotransmitters which would normally be delivered from the higher central nervous system has been shown to produce substantial gains in stepping performance. If made practicable by an incomplete motor injury or some other therapeutic approach, physical training is also very useful for recovering function, as it provides the task-appropriate input to which the spinal cord is being trained to respond appropriately (Wernig and Mller, 1992; Engesser-Cesar et al., 2007), which may induce plastic reorganization of the lower spinal cord. In order to reduce the need for human assistance of the patient during activity-based therapy, a number of efforts have concentrated on robotic locomotor training. Cai et al. (2006) considered how the controller which drives a robotic assistance system can affect the therapeutic outcome, showing that an assist-as-needed paradigm which enforced some interlimb coordination outperformed both rote training of the nominal trajectory and an assist-as-needed controller which did not enforce interlimb coordination. Emken et al. (2008) addressed a similar question of robotic gait training and appropriate control algorithms in humans. The phenomenon of learned helplessness, i.e., non-responsiveness to stimuli which cannot be avoided, is present in the rat spinal cord and can be manipulated by both pharmacology and linkage of lower limb position with noxious stimulus (Crown and Grau, 2001). The results of Cai et al. (2006) may provide evidence that variability in the training paradigm is important for avoiding this outcome.

Epidural Electrical Stimulation

Another important cord-therapeutic technique for SCI therapy, and the focus of the applied portions of this dissertation, is epidural electrical stimulation. While originally developed for chronic pain therapy (Shealy, Mortimer, and Reswick, 1967; Shealy, Taslitz, et al., 1967), spinal electrical stimulation can produce complex motor patterns (Dimitrijevic, Yuri Gerasimenko, and Pinter, 1998). A variety of methods for delivering the electrical stimulus have been suggested, including penetrating microelectrodes. Epidural spinal cord stimulation has been applied with similar results in spinal and decerebrate cats, spinalized rats, and humans, and when properly configured, can produce walking motions (described in the review by Yury Gerasimenko, Roy, and Edgerton (2008). Herman et al. (2002) combined epidural electrical stimulation with partial body weight support exercise training to produce substantial perceived, functional, and metabolic gains in locomotion in an incomplete quadraplegic patient. More recently, Harkema et al. (2011) used epidural electrical stimulation with training and demonstrated substantial gains in a motor-complete patient in a similar setting. This type of stimulation is believed to activate afferent fibers as they enter the spinal cord through the dorsal nerve roots (Minassian et al., 2007).

Combined Approaches

It is often the case that the therapeutic approaches outlined above can be combined for improved effects. For example, Capogrosso et al. (2016) examine serotonin agonists and electrical stimulation for excitation of the spinal cord in treadmill walking, and

Brand et al. (2012) use electrical stimulation, pharmacology, and a compliant robotic assist device, both in SCI rats. Both works show impressive functional gains, with the latter showing a restoration of voluntary locomotion. As mentioned in Edgerton et al. (2006), it remains an open question as to how to optimally combine individual, disparate therapies into a therapeutic program.

2.3 Bandit Learning

This section provides an overview of active learning and bandit algorithms related to the work in this dissertation. Intuitively, a learner which asks useful questions should be able to learn more information and use fewer observations than a learner which waits for informative data to arrive by chance. For a view of the traditional field of active learning, i.e., the field of actively querying algorithms which do not obtain reward or suffer regret, the interested reader may refer to the text by Settles (2012). Since this work combines ideas from bandits and Bayesian optimization, a brief review of the literature in each of these areas is included.

Basic Setting

Exploration-exploitation tradeoffs have been classically studied in the context of the (stochastic) multi-armed bandit problem, in which, from among some finite set of candidate actions, a single action is chosen at each round, and the corresponding (possibly noisy) reward is observed. A recent monograph by Sébastien Bubeck and Cesa-Bianchi (2012) describes a number of related bandit problems and several algorithms for solving each. Briefly, early work has focused on the case of a finite number of decisions and payoffs that are independent across the arms (Robbins, 1952). In this setting, under some strong assumptions, optimal policies can be computed (Lai and Robbins, 1985). Due to the difficulties inherent in doing so, however, a number of heuristic policies have been created. Optimistic allocation of actions according to upper-confidence bounds (UCB) on the payoffs has proven to be particularly effective (Auer, Cesa-Bianchi, and Fischer, 2002).

Bandit Theory

The original i.i.d. multi-armed bandit problem was proposed in Robbins (1952). The problem formulation is reviewed below.

- Known parameters: number of arms K and (possibly) number of rounds $T \geq K$.

- Unknown parameters: K probability distributions $\nu_1, \dots, \nu_K$ on $[0, 1]$ with mean $\mu_1, \dots, \mu_K$ (notation: $\mu^* = \max_{i \in [K]} \mu_i$).
- Protocol: For each round $t = 1, 2, \dots, T$, the player chooses $I_t \in [K]$ based on past observations and receives a reward/observation $Y_t \sim \nu_{I_t}$ (independently from the past).
- Performance measure: The cumulative regret is the difference between the player's accumulated reward and the maximum the player could have obtained had she known all the parameters,

$$\mathbb{E}R_T = T\mu^* - \mathbb{E} \sum_{t \in [T]} Y_t.$$

This problem models the fundamental tension between exploration and exploitation where one wants to pick arms that performed well in the past, yet one needs to make sure that no good option has been missed. More and more applications are found that fit this simple framework, such as advertisement placement on the Internet.

There are fundamental limitations of i.i.d. multi-armed bandit.

First, there exists lower bounds for $\mathbb{E}R_T$. Consider the 2-armed case where $\nu_1 = Ber(1/2)$ and $\nu_2 = Ber(1/2 + \xi\Delta)$ where $\xi \in \{-1, 1\}$ is unknown. Ber denotes the Bernoulli distribution and Δ is the marginal difference between ν_1 and nu_2. With τ expected observations from the second arm there is a probability at least $\exp(-\tau\Delta^2)$ to make the wrong guess on the value of ξ. Now let $\tau(t)$ be the expected number of pulls of arm 2 when $\xi = -1$. One has

$$\mathbb{E}R_T(\xi = +1) + \mathbb{E}R_T(\xi = -1) \geq \tau(T)\Delta + \sum_{t=1}^{T} \exp(-\tau(t)\Delta^2) \geq \min_{t \in [T]}(t\Delta + T\exp(-t\Delta^2)) \approx \frac{\log(T\Delta^2)}{\Delta}.$$

More details can be found in Sébastien Bubeck, Perchet, and Rigollet (2013). The important message is that for Δ fixed the lower bound is $\frac{\log(T)}{\Delta}$. For the worse Δ it is $\sqrt{T}$. In the K-armed case this worst-case lower bound becomes $\sqrt{KT}$. Let $\Delta_i = \mu^* - \mu_i$ and $N_i(t)$ the number of pulls of arm i up to time t. Note that one has $\mathbb{E}R_T = \sum_{i=1}^{K} \Delta_i \mathbb{E}N_i(T)$. For $p, q \in [0, 1]$ let

$$\mathrm{kl}(p, q) := p \log \frac{p}{q} + (1 - p) \log \frac{1 - p}{1 - q}$$

Theorem (Lai and Robbins, 1985) considers a multi-armed bandit strategy s.t. $\forall a > 0$, we have $\mathbb{E}N_i(T) = o(T^a)$ if $\Delta_i > 0$. Then for any Bernoulli distributions,

$$\liminf_{n\to+\infty} \frac{\mathbb{E}R_T}{\log(T)} \geq \sum_{i:\Delta_i>0} \frac{\Delta_i}{\mathrm{kl}(\mu_i, \mu^*)}.$$

Note that $\frac{1}{2\Delta_i} \geq \frac{\Delta_i}{\mathrm{kl}(\mu_i,\mu^*)} \geq \frac{\mu^*(1-\mu^*)}{2\Delta_i}$, so up to a variance-like term the Lai and Robbins lower bound for the multi-armed bandit problem is $\sum_{i:\Delta_i>0} \frac{\log(T)}{2\Delta_i}$.

The UCB (Upper Confidence Bound) strategy (Lai and Robbins, 1985; Auer, Cesa-Bianchi, and Fischer, 2002) is,

$$I_t \in \operatorname*{argmax}_{i\in[K]} \mathrm{UCB}_i(t).$$

The regret analysis is on a $1 - 2/T$ probability event one has

$$N_i(t) \geq 8\log(T)/\Delta_i^2 \Rightarrow \mathrm{UCB}_i(t) < \mu^* \leq \mathrm{UCB}_{i^*}(t),$$

so that $\mathbb{E}N_i(T) \leq 2 + 8\log(T)/\Delta_i^2$ and in fact

$$\mathbb{E}R_T \leq 2 + \sum_{i:\Delta_i>0} \frac{8\log(T)}{\Delta_i}.$$

Bayesian multi-armed bandit Thompson Sampling (TS) (Thompson, 1933)

Assume a set of models $\{(\nu_1(\theta), \ldots, \nu_K(\theta)), \theta \in \Theta\}$ and prior distribution π_0 over Θ. The Bayesian regret is defined as

$$BR_T(\pi_0) = \mathbb{E}_{\theta\sim\pi_0}\mathbb{E}R_T(\nu_1(\theta), \ldots, \nu_K(\theta)),$$

where $\mathbb{E}R_T(\nu)$ simply denotes the regret for the i.i.d. model when the underlying reward distributions are $\nu_1, \ldots, \nu_K$. In principle the strategy minimizing the Bayesian regret can be computed by dynamic programming on the potentially huge state space $P(\Theta)$. The Gittins index theorem (Gittins, Glazebrook, and Weber, 2011) gives sufficient condition to dramatically reduce the computational complexity of implementing the optimal Bayesian strategy under a strong product assumption on π_0. Notation: π_t denotes the posterior distribution on θ at time t.

Theorem (Gittins Index Theorem, Gittins, '74, '79, '89) The expected discounted reward obtained from a simple family of alternative bandit processes is maximized by always continuing the bandit having greatest Gittins index,

$$G_i(x_i) = sup_{\tau \geq 1} \frac{E[\Sigma_{t=0}^{\tau-1} r_i(x_i(t))\beta^t | x_i(0) = x_i]}{E[\Sigma_{t=0}^{\tau-1} \beta^t | x_i(0) = x_i]}$$

where τ is a (past-measurable) stopping time.

Thompson proposed the following strategy: sample $\theta' \sim \pi_t$ and play $I_t \in \text{argmax}\, \mu_i(\theta')$.

Theoretical guarantees for this highly practical strategy have been provided by Agrawal and Goyal (2012) and Kaufmann, Korda, and Rémi Munos (2012). They proved that TS with Bernoulli reward distributions and uniform prior on the parameters achieves $\mathbb{E}R_T = O\left(\sum_i \frac{\log(T)}{\Delta_i}\right)$.

Bandit Algorithms

Many problems have a repeating game structure, in which there exist, a set of alternatives and the agent must choose one among these at each round. The agent then receives the reward corresponding to this action. Crucially, only this (possibly noisy) reward is observed, while rewards corresponding to other actions are unrevealed; this suggests that, in order to obtain a good amount of reward, the agent must use a strategy which exploits knowledge of the reward function to obtain high reward, and explores the reward function thoroughly enough to be assured that the action which is apparently best is, in fact, the one which yields the highest reward. The balance between these competing imperatives is referred to as the exploration-exploitation tradeoff. The most crucial division among algorithms in the bandit class is in regard to the types of structural assumptions made about the reward function, i.e., whether the payoffs corresponding to individual actions are somehow related to one another, or if they are totally independent.

Structural Assumptions for Large Problems

Recently, approaches for coping with large (or infinite) sets of decisions have been developed. In these cases, since the number of candidate actions is very large compared to the number of actions to be allocated, the reward function cannot be adequately learned if the payoffs are independent. In order to achieve some level of tractability, the dependence between the payoffs associated with different candidate actions must be modeled and exploited. Examples include bandits with linear (Dani, Hayes, and Kakade, 2008; Abernethy, Hazan, and Rakhlin, 2008) or

Lipschitz-continous payoffs (Kleinberg, Slivkins, and Upfal, 2008), or bandits on trees (Kocsis and Szepesvári, 2006b; Sébastien Bubeck, Rémi Munos, et al., 2008). Chapter 5 pursues a Bayesian approach to bandits, where fine-grained assumptions on the regularity of the reward function can be imposed through proper choice of the prior distribution.

Bayesian Optimization

The exploration-exploitation tradeoff has also been studied in Bayesian global optimization and response surface modeling, where Gaussian process models are often used due to their flexibility in incorporating prior assumptions about the structure of the payoff function (Brochu, Cora, and Freitas, 2010). Several bandit-like heuristics, such as Maximum Expected Improvement (Jones, Schonlau, and Welch, 1998), Maximum Probability of Improvement (Mockus, 1989), Knowledge Gradient (Ryzhov, Powell, and Frazier, 2012), and upper-confidence based methods (Cox and John, 1997), have been developed to balance exploration with exploitation and have been successfully applied in different learning problems. In contrast, the Entropy Search algorithm of Hennig and Schuler (2012) considers the estimate of the location of the optimum at any given time and tries to take the action which will greedily decrease future losses, a less bandit-like and more optimization-focused heuristic. Srinivas et al. (2010a) analyzed GP-UCB, an upper-confidence bound sampling based algorithm for this setting, and proved bounds on its cumulative regret, and thus convergence rates for Bayesian global optimization. T. Desautels, Krause, and J. Burdick (2012) studied a batched version of the GP-UCB.

Online Learning

Some attempts to use algorithmic methods for managing the interaction of therapeutic systems with complex biological systems have been made in the past. For example Santaniello et al. (2011) studied the simulation of the closed-loop control of deep brain stimulation. They modeled the responses of simulated neurons in the ventral intermediate nucleus of the thalamus to deep brain electrical stimulation as a parametric dynamic model, with coefficients fitted online. They controlled the application of the stimulator to attempt to disrupt tremor-like activity in this population of simulated cells. Another application of interest has been brain-computer interface (BCI). Traditionally, BCI uses fairly simple decoding algorithms, which classify neural activity by comparison to pre-computed, possibly stereotyped patterns corresponding to putative volitional states. Fruitet, Carpentier, Clerc, et al.

(2012) developed and tested in humans a bandit-based algorithm to create personalized BCI (Fruitet, Carpentier, Rémi Munos, et al., 2013). This algorithm chose which action to ask the user to imagine performing adaptively, and eventually produced good discrimination between this neurological state and the resting state, thus creating a classifier for the state of a volitional "button-press" manifested in the patient's sensori-motor rhythms. While intended to ultimately work with much larger sets of imagined motor actions, these experiments used a menu of three to five possible actions.

Gürel and Mehring (2012) used what is essentially an ϵ-greedy bandit as a meta-algorithm for online, continuing calibration of a BCI decoding process, following an initial supervised training stage. Vidaurre et al. (2011) employed a multi-phase calibration of such a system, including the user's immediate feedback responses to the online-decoded intention. The work in SCI therapy uses more structure over the space of actions (over which the reward function is modeled as a Gaussian process or as a correlated dueling bandit problem which is described below) in order to enable the use of a very large decision set.

2.4 Dueling Bandit

The dueling bandits problem is an online learning framework for learning from preference feedback, and is particularly well-suited for modeling settings that elicit subjective or implicit human feedback.

Consider the following sequential optimization problem with relative feedback. Let $\mathcal{B} = \{b_1, \ldots, b_K\}$ be the set of K bandits (or arms). At each round $t = 1, 2, \cdots$, the system presents a pair of arms b_i, b_j from the set of K arms based on users picking criteria (or dueling bandits algorithm). Arms b_i and b_j may be identical. We assume the outcome of each duel or comparison between b_i and b_j is an independent sample of a Bernoulli random variable $X(b_i, b_j)$. We define the probability that arm b_i beats b_j as

$$P(b_i \succ b_j) = \phi(b_i, b_j) + 1/2,$$

where $\phi(b_i, b_j) \in [-1/2, 1/2]$ denotes the underlying preference between b_i and b_j. Obviously, $b_i \succ b_j$ if and only if $\phi(b_i, b_j) > 0$. We also assume that there is a total ordering, and that the bandits are indexed in that ordering WLOG: $b_i \succ b_j \Leftrightarrow i < j$.

The setting proceeds in a sequence of iterations or rounds. At each iteration t, the decision maker must choose a pair of bandits $b_t^{(1)}$ and $b_t^{(2)}$ to compare, and observes the outcome $X(b_t^{(1)}, b_t^{(2)})$ of that comparison. The quality of the decision making is then quantified using a notion of cumulative regret of T iterations

$$R_T = \sum_{t=1}^{T} \left[\phi(b_1, b_t^{(1)}) + \phi(b_1, b_t^{(2)}) \right]. \tag{2.1}$$

To date, there have been several algorithms proposed for the stochastic K-armed dueling bandits problem, including Interleaved Filter (Yue et al., 2012), Beat the Mean (Yue and Joachims, 2011), SAVAGE (Urvoy et al., 2013), RUCB (Zoghi, Whiteson, Remi Munos, et al., 2014; Zoghi, Whiteson, and Rijke, 2015), and Sparring (Ailon, Z. Karnin, and Joachims, 2014; Dudik et al., 2015). Of these, RUCB and Sparring consistently achieve the best empirical performance (Ailon, Z. Karnin, and Joachims, 2014). Chapter 3 introduces a new algorithm that builds upon Sparring to arrive at a new algorithm, which we call SelfSparring, SelfSparring significantly outperforms the state-of-the-art algorithms in empirical evaluations.

2.5 Gaussian Processes

Gaussian processes (GPs) are a flexible model for capturing knowledge about functions from a variety of classes. Rasmussen and Williams (2006) provide an excellent introduction to GPs. This section presents a brief review of relevant parts of GP theory and practice.

Rasmussen and Williams (2006) define a Gaussian process as follows: a Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution.

Another way of thinking of a Gaussian processes is to describe it as a probability distribution over functions mapping from an arbitrary (possibly continuous) index set S to R. To denote that such a function $f : D \Rightarrow \mathbb{R}$ is drawn from a GP over functions on D, one may write $f \sim GP(\mu(x), k(x, x'))$, where $x, x' \in D$, $\mu(x)$ is the mean function and $k(x, x')$ is the covariance function. Any element $x \in D$ corresponds to the identity of a random variable, and the value of any function drawn from the GP at x, $f(x)$, corresponds to a particular assignment of a value to the random variable identified by x. Note that here and in the remainder of the text, we use the notation f to denote a function over D and $f(\cdot)$ to denote the value of that function at a finite collection of elements in D. A GP is fully specified by the mean function and covariance function; for any collection of elements of the GP, these may be used to define the Gaussian joint distribution over the values of those random variables. For example, on any collection of $n \in \mathbb{N}^+$ elements of D, where this collection of points is described as a column vector, $X = [x_1, ..., x_t]^T$, the Gaussian joint distribution over this column vector of corresponding values of f is $f(X) = [f(x_1), ..., f(x_t)]^T \sim N(\mu(X), K(X, X))$, where $\mu(X)$ is the column vector of values of the mean function and $K(X, X)$ is the covariance matrix, and where the entries of $K(X, X)$ are $[K(X, X)]_{ij} = k(x_i, x_j), \forall i, j \leq t$.

In particular, for any $x, x' \in D$, the covariance of $f(x)$ and $f(x')$ is $k(x, x')$. The existence of a covariance function automatically grants a consistency property, which is the distribution of any sub-collection of random variables from the GP.

This dissertation will assume without loss of generality that the prior for mean function $\mu(x)$ is zero everywhere in D. It is mathematically equivalent to perform regression on deviations from $\mu(x)$, expressed as $f(x) - \mu(x)$, rather than on the actual value of the function $f(x)$, and thus the corresponding change of definitions is preferred for simplicity of presentation and calculation.

Regression Using Gaussian Processes

Chapter 5 of this work considers using the GP model to make predictions about $f(x)$, the value of a function drawn from the GP at a test point x, given some finite set of observations y corresponding to the set X. Assuming i.i.d. Gaussian noise on these observations with noise variance σ_n^2, and denoting the size of X as t, the individual observations corresponding to $x_i \in X$ may be written as $y_i = f(x_i) + \epsilon_i$, $i \in 1, ..., t$, where $\epsilon_i \sim N(0, \sigma_n^2)$, $\forall i \in 1, ..., t$. The joint distribution over the observations $y = [y_1, ..., y_t]^T$ This assumption allows us to model our reward function f as a sample from a Gaussian process (GP) (Rasmussen and Williams, 2006). A $GP(\mu(\times x), k(x, x'))$ is a probability distribution across a class of "smooth" functions, which is parameterized by a kernel function $k(x, x')$ that characterizes the smoothness of f. We assume w.l.o.g. that $\mu(x) = 0$, and that our observations are perturbed by i.i.d. Gaussian noise. At points $A_T = [x_1 \dots x_T]^T \subseteq D$, we have $y_t = f(x_t) + n_t$ where n_t $N(0, \sigma^2)$. (We will relax this assumption later.) The posterior over f is then also Gaussian with mean $\mu_T(x)$, covariance $k_T(x, x')$ and variance $\sigma_T^2(x, x')$ that satisfy,

$$\begin{aligned}
\mu_T(x) &= k_T(x)^T (K_T + \sigma^2 I)^{-1} y_T \\
k_T(x, x') &= k(x, x') - k_T(x)^T (K_T + \sigma^2 I)^{-1} k_T(x') \\
\sigma_T^2(x) &= k_T(x, x),
\end{aligned}$$

where $k_T(x) = [k(x_1, x) \dots k(x_T, x)]^T$ and K_T is the positive definite kernel matrix $[k(x, x')]_{x, x' \in A_T}$.

These forms represent the uncertainty over which function from the Gaussian process explains the observations, and capture the marginalization over all functions which could be drawn from the GP; this implicit marginalization is a manifestation of the consistency property.

2.6 Covariance Functions

The previous discussion has assumed the availability of a covariance function (kernel function) $k(\cdot, \cdot)$, but covariance functions are themselves a topic of significant interest. In applications, the choice of covariance function is a major opportunity to specify the structure of the problem, as expert knowledge can be used to choose a covariance function which encodes a great deal of problem-specific knowledge. Such choices result in relatively stronger or weaker links between the values of f at various pairs of elements x, x' from within the chosen domain S of the covariance function (which is the domain of functions drawn from the corresponding GP). Further, some regions of S could be specified to have larger variances than others, encoding the knowledge that some particular region of the space is known to produce more variable behavior. Similarly, in $\mathbb{R}_d$, a covariance function could be constructed to produce draws from the GP which vary more slowly in certain directions than others; this can be very useful, e.g., if the system being modeled is known to be relatively insensitive to one of the variables describing the location in $\mathbb{R}_d$, whereas it is more sensitive to others. As covariance functions are a crucial topic in understanding GPs, Rasmussen and Williams (2006) also provide a thorough description of this topic. A brief introduction of the relevant details is presented here. A real kernel function k is a function which maps pairs of elements of D into R, i.e., $k : D \times D \rightarrow \mathbb{R}$. A kernel which is symmetric in its arguments, i.e., $k(x, x') = k(x', x)$, is referred to as a symmetric kernel. In Euclidean spaces, the stationary and isotropic properties are of interest. If k is solely a function of $x - x'$, k is stationary, and it is invariant to translations of the inputs. If k is a function of only the (vector) magnitude of this difference, $|x - x'|$, it is isotropic. Covariance functions are a sub-class of symmetric kernel functions. For any kernel function k and collection $X = x_1, \cdots, x_n$ of n elements of D, the Gram matrix is $K(X, X)$, where $[K(X, X)]_{ij} = k(x_i, x_j)$, $\forall i, j \leq n$. If for a symmetric kernel k, the Gram matrix is positive semi-definite $\forall n \in \mathbb{N}^+$, k is termed positive semi-definite. If a kernel k is symmetric and positive semi-definite, k is also a valid covariance function, and any Gram matrix corresponding to k is referred to as a covariance matrix. The covariance function in essence defines similarity between the values of

$f(x)$ and $f(x')$ for any two elements $x, x' \in D$, and does so by reference to x and x', rather than the function itself. This is useful for a variety of reasons:

- For a finite collection of points $d \in D$, the covariance matrix $K(d, d)$ of the jointly Gaussian values of f at the elements of d can be computed before hand. Conditioned on observations of elements of d, the posterior over f at d can be computed using this matrix.

- Complex representations of x and x' in feature spaces, even infinite-dimensional feature spaces, can be encoded by using a kernel (covariance) function which operates (typically very simply) on x and x'; this is commonly known in machine learning as the kernel trick, and is employed to leverage a simple technique into a much more complex and expressive suite of techniques while incurring very little computational expense. In this sense, Gaussian process regression is actually Bayesian linear regression, extended via the kernel trick.

- From a practical perspective, careful choices of the representation of the inputs, the kernel function, and the corresponding hyperparameters can allow expert knowledge to be encoded into the GP model very simply and intuitively. As an expert works with a system, they might plausibly acquire some intuition of which variables are functionally important and which are less so. It might be that there are many ways to describe the objects in the input space, but some may be more convenient or meaningful than others; in essence, this is an extension of the kernel trick. The choice of covariance function also is an opportunity for expert intuition to be expressed; linear or squared-exponential kernels imply quite different things about the functions drawn from the corresponding GPs. Similarly, the choice of the hyperparameters of selected kernel function encodes information like the relative sensitivity of the responses to variation in any of the chosen features.

Reproducing Kernel Hilbert Spaces

A reproducing kernel Hilbert space (RKHS) is a Hilbert space of functions over a set D associated with a particular kernel function. More precisely, an RKHS is defined by Rasmussen and Williams (2006) as follows: Let H be a Hilbert space of real functions f defined on an index set D. Then H is called a reproducing kernel Hilbert space (RKHS) endowed with an inner product $< \cdot, \cdot >_H$ (and norm $||f||_H =< f, f >_H$). If there exists a function $k : D \times D \to \mathbb{R}$ with the following properties:

- for every x, $k(x, x')$ as a function of x belongs to H.
- k has the reproducing property $< f(\cdot), k(\cdot, x) >_H = f(x)$.

For the purposes of structural learning of input space, it is most important to note that the RKHS norm of a function f provides a measure of how closely f matches the possible posterior means, which would be constructed from a GP model using the corresponding kernel and a finite amount of data, i.e., how well f can be captured by the model; a small value for $||f||_H$ implies that f is much like a linear combination of relatively few copies of the kernel function, whereas a large or infinite value for $||f||_H$ implies that this is not the case. Alternatively, a finite value of $||f||_H$ could be viewed as the rapid decay of the eigenvalues of f with respect to an eigenfunction basis of the RKHS.

Constructing Covariance Functions

It is also possible to construct more complicated covariance functions by using compositions of simpler covariance functions. In particular, the sum of two covariance functions is a covariance function, and a sample from the GP corresponding to this covariance function corresponds to a sum of independent samples from the two GPs which correspond to the two original covariance functions. Similarly, the product of two covariance functions is also a covariance function, such that draws from the GP corresponding to the product covariance function can be thought of as being the product of two independent draws from the GPs corresponding to the individual factor covariance functions. Finally, two covariance functions $k_1(x_1, x'_1)$ and $k_2(x_2, x'_2)$ over different spaces D_1 and D_2 may be combined as either a sum $k(x, x') = k_1(x_1, x'_1) + k_2(x_2, x'_2)$ or product $k(x, x') = k_1(x_1, x'_1) \times k_2(x_2, x'_2)$ to form a covariance function k for $x, x' \in D_1 \times D_2$ via the sum and product methods above. This allows the construction of covariance functions from individual covariance functions over subspaces, e.g., different dimensions of $\mathbb{R}_d$, or even radically different sets; D_1 might be $\mathbb{R}$, whereas D_2 could be nodes in a graph, words in a corpus, or something more exotic. While it is possible to construct covariance functions which natively represent covariance over a space D which is not a subset of $\mathbb{R}_d$, another way to construct covariance functions for such D is to find a mapping: $D \longrightarrow \mathbb{R}_d$ and then use a covariance function $k \sim: \mathbb{R}_d \times \mathbb{R}_d \Rightarrow \mathbb{R}$ to construct a covariance function, such that $k(x, x') = k \sim (g(x), g(x'))$. The combination of all of these techniques allows a great deal of flexibility in terms of modeling assumptions.

Chapter 3

THEORETICAL CONTRIBUTIONS: ALGORITHMS FOR DUELING BANDITS

This chapter provides several new algorithms for dueling bandits problem, a variant of the multi-armed bandit problem. The dueling bandits problem is an online learning framework for learning from preference feedback, and is particularly well-suited for modeling settings that elicit subjective or implicit human feedback. This chapter shows how to view the dueling bandits problem as a two-player game with stochastic rewards and slowly drifting dynamics. This chapter also studies the problem of *multi-dueling bandits with dependent arms*, which extends the original dueling bandits setting by simultaneously dueling multiple arms as well as modeling dependencies between arms. These extensions capture key characteristics found in many real-world applications, and allow for the opportunity to develop significantly more efficient algorithms than were possible in the original setting.

3.1 Introduction

In many online learning settings, particularly those that involve human feedback, reliable feedback is often limited to pairwise preferences (e.g., "is A better than B?"). Examples include implicit or subjective feedback for information retrieval and recommender systems (e.g., clicks on search results, or subjective feedback on the quality of recommended care) (Chapelle, Joachims, et al., 2012; Sui and J. Burdick, 2014b). This setup motivates the dueling bandits problem (Yue et al., 2012), which formalizes the problem of online regret minimization via preference feedback (e.g., choosing a pair of arms to be compared at each time step).

One of the best performing algorithms is the Sparring algorithm (Ailon, Z. Karnin, and Joachims, 2014), which uses two separate multi-armed bandit algorithms to choose the two arms to be compared at each time step, and essentially treats the dueling bandits problem as a competition between two learning agents. Operationally, Sparring uses two separate multi-armed bandit algorithms to choose the two arms to be compared at each time step. This viewpoint reveals a close connection between the dueling bandits problem and online learning in two-player zero-sum games. As such, designing efficient no-regret dueling bandit algorithms is closely related to designing online learning agents that quickly converge to the Nash equilibrium in the corresponding two-player game. Furthermore, if there is a Condorcet winner [1] in the arms (i.e., a dominating strategy), then the optimal solution to the Dueling Bandits problem is also a unique pure Nash equilibrium in a corresponding two-player zero-sum game.

This chapter shows how to view the dueling bandits problem as a two-player game with stochastic rewards and slowly drifting dynamics. Through this viewpoint, I provide the first near-optimal no-regret guarantee for a variant of Sparring using a stochastic bandit algorithm, which we call SelfSparring. One important property that we leverage is *approximate linearity*, which fully generalizes the linear utility-based dueling bandits setting studied in Ailon, Z. Karnin, and Joachims (2014) – see Section 3.6 for more details. We also demonstrate empirically that SelfSparring achieves state-of-the-art performance.

3.2 Problem Setups

Consider the following sequential optimization problem with relative feedback. Let $\mathcal{B} = \{b_1, \ldots, b_K\}$ be the set of K arms. At each round $t = 1, 2, \cdots$, the system presents a pair of arms b_i, b_j from the set of K arms based on users picking criteria (or dueling bandits algorithm). b_i and b_j can be identical. We assume the outcome of each duel or comparison between b_i and b_j is an independent sample of a Bernoulli random variable $X(b_i, b_j)$. We define the probability that arm b_i beats b_j as

$$P(b_i \succ b_j) = \phi(b_i, b_j) + 1/2,$$

where $\phi(b_i, b_j) \in [-1/2, 1/2]$ denotes the preference between b_i and b_j. Obviously, $b_i \succ b_j$ if and only if $\phi(b_i, b_j) > 0$. We also assume that there is a total ordering, and that the bandits are indexed in that ordering WLOG: $b_i \succ b_j \Leftrightarrow i < j$.

The setting proceeds in a sequence of iterations or rounds. At each iteration t, the decision maker must choose a pair of bandits $b_t^{(1)}$ and $b_t^{(2)}$ to compare, and observes the outcome $X(b_t^{(1)}, b_t^{(2)})$ of that comparison. The quality of the decision making is then quantified using a notion of cumulative regret of T iterations:

$$R_T = \sum_{t=1}^{T} \left[\phi(b_1, b_t^{(1)}) + \phi(b_1, b_t^{(2)}) \right]. \tag{3.1}$$

To date, there have been several algorithms proposed for the stochastic K-armed dueling bandits problem, including Interleaved Filter (Yue et al., 2012), Beat the

[1] The Condorcet winner is the person who would win a two-candidate election against each of the other candidates in a plurality vote. For a set of candidates, the Condorcet winner is always the same regardless of the voting system in question.

Table 3.1: Example $\textsc{Prob}(Row > Col) - 1/2$ preference matrix.

	A	B	C	D	E	F
A	0	0.04	0.05	0.07	0.10	0.12
B	-0.04	0	0.04	0.06	0.08	0.10
C	-0.05	-0.04	0	0.01	0.04	0.09
D	-0.07	-0.06	-0.01	0	0.02	0.05
E	-0.10	-0.08	-0.04	-0.02	0	0.03
F	-0.12	-0.10	-0.09	-0.05	-0.03	0

Mean (Yue and Joachims, 2011), SAVAGE (Urvoy et al., 2013), RUCB (Zoghi, Whiteson, Remi Munos, et al., 2014; Zoghi, Whiteson, and Rijke, 2015), and Sparring (Ailon, Z. Karnin, and Joachims, 2014; Dudik et al., 2015). Of these, RUCB and Sparring consistently achieve the best empirical performance (Ailon, Z. Karnin, and Joachims, 2014). Our work builds upon Sparring to arrive at a new algorithm, called SelfSparring, that significantly outperforms Sparring in empirical evaluations and enjoys asymptotic no-regret guarantees.

SelfSparring relies on the following assumption:

Approximate Linearity: For any triplet of bandits $b_i \succ b_j \succ b_k$ and some constant $\gamma > 0$:

$$\phi(b_i, b_k) - \phi(b_j, b_k) \geq \gamma \phi(b_i, b_j). \tag{3.2}$$

Illustration of a special case of Approximate Linearity: To visualize Approximate Linearity we consider the special case when the preference function $\phi(b_i, b_j) = \Phi(u_i - u_j)$ holds for all bandit pairs (i, j). u_i is a bounded measure of the utility of playing arm b_i. In this case, the approximate linearity of $\phi(\cdot, \cdot)$ is equivalent to having the function $\Phi(\cdot)$ not deviating much from some linear function on its bounded support, as shown in Figure 3.1. Also, any monotonic increasing function $\Phi(\cdot)$ satisfies approximate linearity. When Φ is linear, then the problem setting reduces to the utility-based dueling bandits setting studied in Ailon, Z. Karnin, and Joachims (2014).[2]

[2] Compared to the assumptions of the original dueling bandits setting (Yue et al., 2012), one can show that Approximate Linearity is a stricter requirement than strong stochastic transitivity, and is a complementary requirement to stochastic triangle inequality. In particular, stochastic triangle inequality requires that the curve in Figure 3.1 exhibits diminishing returns in the top-right quadrant (i.e., is sub-linear), whereas Approximate Linearity requires that the curve be not too far from linear.

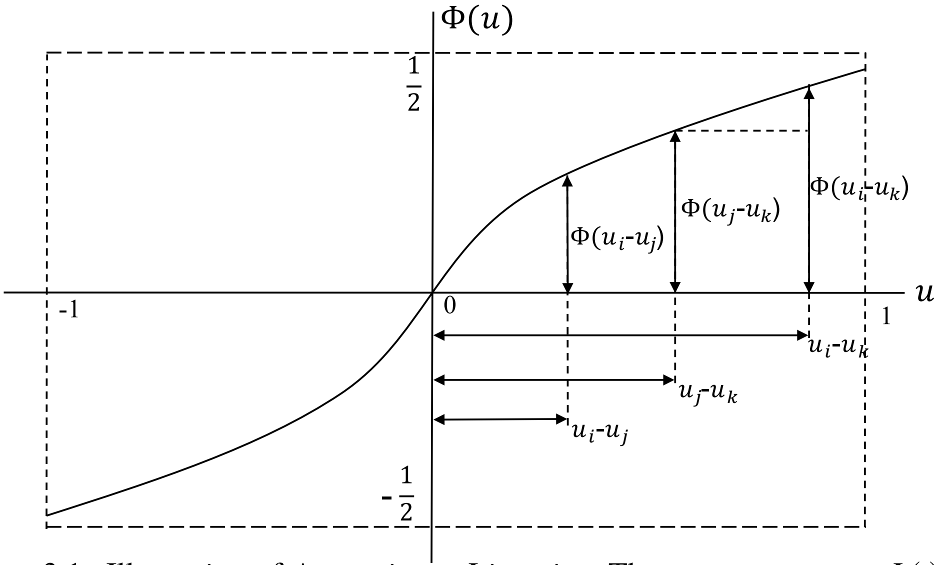


Figure 3.1: Illustration of Approximate Linearity. The curve represents $\Phi(\cdot)$ with support on $[-1, 1]$. The strictly increasing $\Phi(\cdot)$ guarantees the positive value of $\Phi(u_i - u_k) - \Phi(u_j - u_k)$. Then approximate linearity holds for some γ.

3.3 Background

Multi-Armed Bandits

Both Sparring and SelfSparring utilize a multi-armed bandit (MAB) algorithm as a subroutine, and so we provide here a brief formal description of the MAB problem for completeness. The stochastic MAB problem (Robbins, 1952) refers to an iterative decision making problem in which we repeatedly choose among K options, such as pulling one of K arms of a bandit machine. In each round, we receive a reward that depends on the arm being selected. Without loss of generality, assume that every reward is bounded between [0, 1].[3] The goal then is to minimize the cumulative regret compared to the best arm:·

$$R_T^{\text{MAB}} = \sum_{t=1}^{T} [\mu^* - \mu(b_t)], \tag{3.3}$$

where b_t denotes the arm chosen at time t, $\mu(b)$ denotes the expected reward of arm b, and $\mu^* = \text{argmax}_b\, \mu(b)$. Popular algorithms for the stochastic setting include UCB (upper confidence bound) algorithms (Auer, Cesa-Bianchi, and Fischer, 2002; Sébastien Bubeck and Cesa-Bianchi, 2012), and Thompson Sampling (Chapelle and Li, 2011; Russo and Van Roy, 2014).

[3] So long as the rewards are bounded, one can shift and re-scale them to fit within [0, 1].

In the adversarial setting, the rewards are chosen in an adversarial fashion, rather than sampled independently from some underlying distribution. In this case, regret (3.3) is rephrased as the difference in the sum of rewards. The predominant algorithm for the adversarial setting is EXP3 (Auer, Cesa-Bianchi, Freund, et al., 2002).

All of the algorithms we study in this chapter can, in principle, use any of the previously mentioned MAB algorithms. Previous work showed that Sparring enjoys state-of-the-art empirical performance using UCB1 (Ailon, Z. Karnin, and Joachims, 2014), and near-optimal no-regret guarantees using EXP3 (Dudik et al., 2015) (but with much worse empirical performance).

Two-Player Games

As mentioned earlier, the dueling bandits problem bears a close affinity to learning in two-player zero-sum games. In Table 3.1, if we view each dimension of the matrix as the behavior of a separate agent, then one can view the preference matrix itself as the payoff matrix for the row agent. Since there is exactly one winner and one loser, the payoff matrix for the column agent is exactly the negation of the row agent, hence a zero-sum game. Furthermore, Table 3.1 has a unique (pure) Nash equilibrium of both agents choosing the Condorcet winner (arm A).

When Table 3.1 is unknown a priori, one can formulate the one-sided online learning problem for each agent as a MAB problem, with regret defined relative to the reward obtained from the Nash equilibrum solution. It is known that no-regret learning agents in a two-player zero-sum game are guaranteed to converge to a Nash equilibrium (Friedman and Shenker, 1998). Furthermore, in the case where there is a Condorcet winner (as is the case in our setting), the Nash equilibrium is a pure Nash, i.e., a deterministic strategy (e.g., arm A in Table 3.1). Thus, one can directly relate no-regret learning for the dueling bandits problem to efficient convergence to the Nash equilibrium in the corresponding two-player game. Indeed, the analysis of Sparring using EXP3 (Dudik et al., 2015) almost exactly follows the analysis of how no-regret online learning converges to a Nash equilibrium in zero-sum games. More generally, the study of learning in games is an area of intense interest within the machine learning and algorithmic game theory communities (Syrgkanis et al., 2015).

The main difficulty in analyzing Sparring using stochastic bandit algorithms such as UCB1 and Thompson Sampling is the fact that the “environment” from the perspective of each learning agent is not static, but rather drifts with the decision

Algorithm 1 Thompson Sampling Subroutines

```
function init()
    Store $D \equiv \{S_1, \cdots, S_K; F_1, \cdots, F_K\} \leftarrow \{\mathbf{0}\}$
end function
function recalibrate($L_n$, $L_{n-1}$)
    for $i$ in $1, \ldots, K$ do
        $S_i \leftarrow S_i \cdot \frac{P_{L_n}(i)}{P_{L_{n-1}}(i)}$
        $F_i \leftarrow F_i \cdot \frac{P_{L_n}(i)}{P_{L_{n-1}}(i)}$
    end for
end function
function queryAction()
    For each arm $i = 1, 2, \cdots, K$:
    sample $\theta_i$ from $Beta(S_i + 1, F_i + 1)$
    return $i = \arg\max_i \theta_i$
end function
function feedback($i, r, L, G$)
    if $L$ not defined then
        $L \leftarrow$ self.$D$
    end if
    if $G$ not defined then
        $G \leftarrow$ self.$D$
    end if
    if $r = 1$ then
        $S_i \leftarrow S_i + \frac{P_L(i)}{P_G(i)}$
    else
        $F_i \leftarrow F_i + \frac{P_L(i)}{P_G(i)}$
    end if
end function
```

making of the other agent. We will show how to incorporate importance weighting into Thompson Sampling to calibrate the drifting environments against a single reference environment to prove no-regret guarantees.

Thompson Sampling

Thompson Sampling is a stochastic MAB algorithm that maintains a distribution over the arms, and chooses arms by sampling from this distribution (Chapelle and Li, 2011; Russo and Van Roy, 2014). This distribution is then updated as feedback is incorporated. The entropy of the distribution thus corresponds to the uncertainty regarding which is the best arm, and flatter distributions lead to more exploration.

Our algorithms rely on Thompson Sampling as a subroutine; hence we define a

Algorithm 2 Thompson Sampling for Bernoulli Bandits

```
$TS \leftarrow$ new Thompson Sampling SBM
$TS$.**init**()
**for** $t = 1, 2, \cdots$ **do**
    $i \leftarrow TS$.**queryAction**()
    Play arm $i$, observe reward $r$
    $TS$.**feedback**($i, r$)
**end for**
```

Thompson Sampling Singleton Bandit Machine (SBM) in Algorithm 1. Let S_i and F_i denote the historical number of wins and losses of arm i, and let D_t denote the set of all parameters at round t:

$$D_t = \{S_1, \cdots, S_K; F_1, \cdots, F_K\}.$$

For brevity, we often represent D_t by D since we only need to keep track of the current distribution. The sampling process of Beta-Bernoulli Thompson Sampling given D is:

- For each arm i, sample $\theta_i \sim Beta(S_i, F_i)$.
- Choose the arm with maximal θ_i.

In other words, we model the average utility of each arm using a Beta prior, and rewards for arm i as Bernoulli distributed according to latent mean utility θ_i. As we observe more rewards, we can compute the posterior, which is also Beta distributed by conjugation between Beta and Bernoulli. The sampling process above can be shown to be sampling for the following distribution:

$$P(i|D) = P(i = \operatorname*{argmax}_b \theta_b | D). \tag{3.4}$$

In other words, the probability we choose arm i is equal to the probability that it has the maximal expected reward under the Beta posterior (after observing D).

Algorithm 1 describes the relevant components of the version of Thompson Sampling we use for SelfSparring-IW and SelfSparring, and Algorithm 2 shows how to use these subroutines for the conventional MAB problem. For the basic MAB problem, the components used are **init()**, **queryAction()**, and **feedback(*i*,*r*)**. We discuss **recalibrate(**L_n**,**L_{n-1}**)** later in the subsection of importance weighting.

init(): initializes the observation set D to be empty.

queryAction(): samples from the K arms according to the Beta posterior induced by D.

feedback(i, r): takes arm i, reward r, and updates D. Note that the full definition of **feedback** is more complicated due to incorporating importance weighting. In the standard MAB setting, we simply increment S_i by 1 if $r = 1$ and F_i by 1 if $r = 0$.

We revisit the Thompson sampling method for Bernoulli bandits as shown in Algorithm 2. (Descriptions for the modular functions are at the end of this section.) The algorithm for Bernoulli bandits maintains Bayesian priors on the Bernoulli means μ_i's. Beta distribution is the conjugate prior for Bernoulli rewards. It forms a family of continuous probability distributions on the interval $(0, 1)$. The pdf of the beta distributions, $Beta(\alpha, \beta)$ with parameters $\alpha > 0$, $\beta > 0$, is given by $f(x; \alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}$. The mean of $Beta(\alpha, \beta)$ is $\frac{\alpha}{\alpha+\beta}$. Larger α and β leads to tighter concentration of $Beta(\alpha, \beta)$ around the mean. If the prior is a $Beta(\alpha, \beta)$ distribution, then after observing a Bernoulli trial, the posterior distribution is either $Beta(\alpha + 1, \beta)$ or $Beta(\alpha, \beta + 1)$, depending on whether the trial resulted in a success or failure, respectively. The Thompson Sampling algorithm initially assumes arm i to have prior $Beta(1, 1)$ on μ_i, which is natural because $Beta(1, 1)$ is the uniform distribution on $(0, 1)$. At time t, having observed $S_{i(t)}$ successes (reward = 1) and $F_{i(t)}$ failures (reward = 0) in $k_{i(t)} = S_{i(t)} + F_{i(t)}$ plays of arm i, the algorithm updates the distribution on μ_i as $Beta(S_{i(t)} + 1, F_{i(t)} + 1)$. The algorithm then samples from these posterior distributions of the μ_i's, and plays an arm according to the probability of its mean being the largest.

Thompson Sampling enjoys near-optimal no-regret guarantees, as give by the following lemma (which is a direct consequence of the main theorems in Agrawal and Goyal (2012) and Kaufmann, Korda, and Rémi Munos (2012)). In fact, one can show that any Thompson Sampling algorithm can match the regret of its analogous UCB style algorithm (Russo and Van Roy, 2014).

Lemma 1 *For the K-armed stochastic MAB problem, Thompson Sampling has expected regret:* $\mathbb{E}[R(T)] = O\left(\frac{K}{\Delta} \ln T\right)$*, where* Δ *is the difference between rewards of the best 2 arms and* T *is the total number of rounds.*

Although the empirical performance of Thompson sampling is not necessarily better than that of UCB methods, it is shown below that its sampling process enables the sharing of information between two players (singleton bandit machines).

Algorithm 3 SelfSparring-IW

```
Input: $\omega$
$TS \leftarrow$ new Thompson Sampling SBM
$TS$.**init**()
$L_0 \leftarrow TS.D$
$n \leftarrow 0, t \leftarrow 0$
**while** $t \geq 0$ **do**
    $L_n \leftarrow TS.D$
    $\omega_n \leftarrow \lfloor \omega^n \rfloor$
    $TS$.**recalibrate**($L_n$, $L_{n-1}$)
    **for** $\tau = 1, \ldots, \omega_n$ **do**
        $i_1 = TS$.**queryAction**()
        $i_2 = TS$.**queryAction**()
        play $i_1, i_2$, observe rewards $r_1, r_2$
        $TS$.**feedback**($i_1$, $r_1$, $L_n$)
        $TS$.**feedback**($i_2$, $r_2$, $L_n$)
        $t \leftarrow t + 1$
    **end for**
    $n \leftarrow n + 1$
**end while**
```

Importance weighting. Thompson Sampling is used as the base MAB algorithm is due to the fact that it is probabilistic. SelfSparring-IW uses a variant of Thompson Sampling with importance weighting. Importance weighting is a general technique for estimating properties of a particular distribution of interest, while only having samples generated from a different distribution, and will be used in SelfSparring-IW to control for the drifting distribution induced by the dueling bandits problem.

In Algorithm 2, for importance weighting, **feedback** with the full argument list, and also periodically use **recalibrate** as well.

feedback(i, r, L, G) takes arm i, reward r, reference distribution L, and historical data distribution G as inputs. It updates D by importance sampling L against G. By adding $\frac{P_L(i)}{P_G(i)}$ instead of 1 to the Beta parameters, arm i can be regarded as sampling from L instead of G. In other words, our goal is to calibrate against distribution L when the rewards are coming from distribution G (that is drifting slowly away from L). In the classical Thompson Sampling approach, $G = L$.

recalibrate(L_n, L_{n-1}) updates the recalibration distribution from L_{n-1} to L_n, which requires reweighting D according to the current fixed distribution. This function is only called periodically by SelfSparring-IW.

Algorithm 4 SelfSparring

```
$TS \leftarrow$ new Thompson Sampling SBM
$TS$.**init**(), $t \leftarrow 0$
**while** $t \geq 0$ **do**
    $i_1 = TS$.**queryAction**()
    $i_2 = TS$.**queryAction**()
    play $i_1, i_2$, observe rewards $r_1, r_2$
    $TS$.**feedback**($i_1$, $r_1$)
    $TS$.**feedback**($i_2$, $r_2$)
    $t \leftarrow t + 1$
**end while**
```

3.4 SelfSparring-IW Algorithms

At each round, most existing dueling bandits algorithm would pick one arm (say, the left arm) as a reference and the other arm (the right arm) for exploration/exploitation. If the left arm remains unchanged throughout the game, the right arm is playing against a fixed (stochastic) environment and receiving feedback from it. In this sense, the right arm can be regarded as a Bernoulli bandit. If we only consider the regret of right arm, the following lemma holds.

Lemma 2 *If approximate linearity holds, competing with a fixed distribution of arms leads to one-side optimal regret for any optimal MAB algorithms.*

Although the regret for the reference (left) arm is not guaranteed to be optimal in this setting, the regret of right arm is optimal. Lemma 6 motivates the idea of keeping a virtually fixed reference and let both arms play against it. We implement it in our proposed algorithms, SelfSparring-IW (SelfSparring with Importance-Weighting) and SelfSparring, for the dueling bandits problem.

SelfSparring-IW and SelfSparring. Here, we describe the operation of SelfSparring-IW and SelfSparring, which are outlined in Algorithm 3 and Algorithm 5 respectively. We make extensive use of the Thompson sampling SBM's modular functions defined in Algorithm 1.

SelfSparring-IW uses the framework of the Sparring algorithm proposed by Ailon, Z. Karnin, and Joachims (2014). Sparring allows two SBMs to compete against each other, treating both arms as equals. By contrast, our algorithms draw both arms from just one SBM, hence the name "Self-Sparring." Instead of the more common class of UCB methods, we use Thompson sampling as the SBM, which allows for the sharing of information between arms.

Since the behavior of both arms are non-stationary, we build a virtual reference via importance weighting. This approach leads to SelfSparring-IW. SelfSparring is also shown as a natural extension of SelfSparring-IW in which no recalibration or importance-weighting is performed. In fact, the $\omega = 1$ case of SelfSparring-IW is just SelfSparring with recalibration and one-step importance weighting at every iteration.

SelfSparring iteratively samples two arms from the current Thompson Sampling distribution (lines 11, 12) and competes both against each other. SelfSparring-IW operates similarly, but updates D in a different manner and proceeds in *epochs*. The length of each epoch is controlled by a scaling parameter ω, which sets the length of the nth epoch to be $\omega_n := \lfloor \omega^n \rfloor$ (line 8).

SelfSparring-IW relies on importance-weighting in order to calibrate the information gained during each epoch against a fixed set of distributions L_n. In effect, this allows us to assume that the arms we compete against in epoch n are sampled from the fixed L_n.

L_0 and L_{-1} are both initialized to D_0. From epoch $n-1$ to epoch n, the reference distribution is first set to the current D, then recalibrated by function **recalibrate**(L_n, L_{n-1}), which adjusts the information gained in past iterations against the current reference set of distributions L_n.

Within epoch n, the Beta distribution of arm i is updated with the importance-weighted parameter $\frac{P_{Ln}(i)}{P_{D_t}(i)}$. $P_{L_n}(i)$ is the probability of sampling arm i given the set L_n of Beta distributions by equation (3.4).

Lemma 3 *If approximate linearity holds, competing with a drifting but converging distribution of arms guarantees convergence for Thompson Sampling.*

3.5 Dueling Experiments

We empirically evaluate the performances of SelfSparring-IW and SelfSparring against several other dueling bandit algorithms, including:

- **Interleaved Filter (IF)** Yue et al. (2012)
- **Beat the Mean (BTM)** Yue and Joachims (2011)
- **RUCB** Zoghi, Whiteson, Remi Munos, et al. (2014)

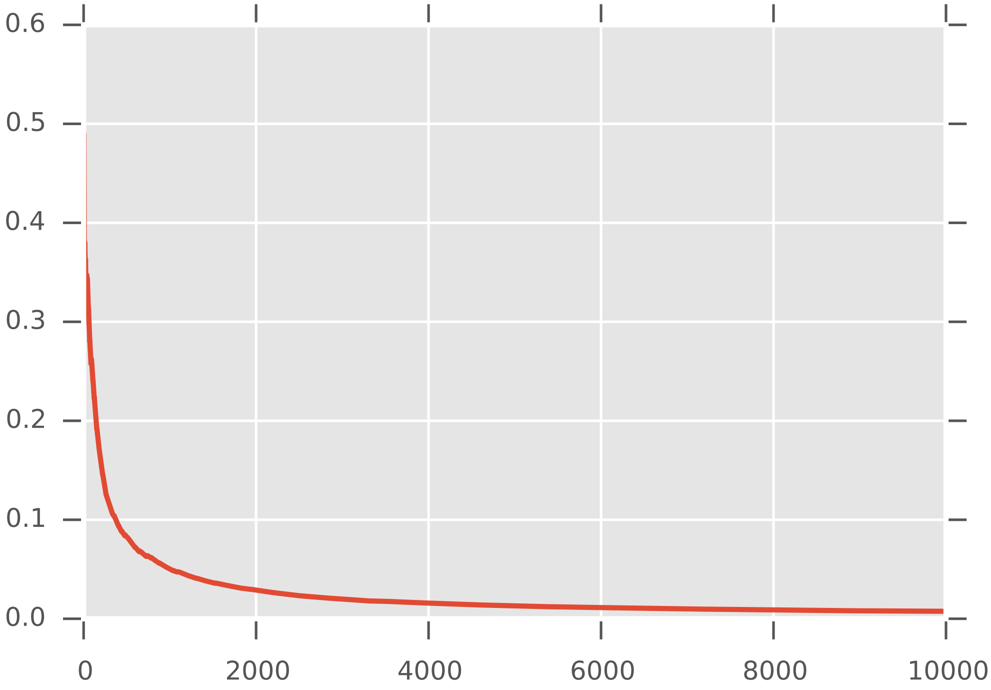


Figure 3.2: Bias ($\mu^* - \bar{\mu}_t$) vs. iterations

- **MergeRUCB** Zoghi, Whiteson, and Rijke (2015)
- **Sparring + UCB1** Ailon, Z. Karnin, and Joachims (2014)
- **Sparring + EXP3** Dudik et al. (2015)

We test these algorithms on 15 six-arm (with arms indexed by {A,B,C,D,E,F}) synthetic scenarios, generated from the three preference functions and five utility functions used in Ailon, Z. Karnin, and Joachims (2014):

$$\begin{array}{ll} \text{linear:} & \phi(x,y) - 1/2 = (1 + x - y)/2 \\ \text{natural:} & \phi(x,y) - 1/2 = x/(x+y) \\ \text{logit:} & \phi(x,y) - 1/2 = (1 + \exp{(y - x)})^{-1} \end{array}$$

with the following utilities:

Name	$\mu(A)$	$\mu(B)$	$\mu(C)$	$\mu(D)$	$\mu(E)$	$\mu(F)$
1good	0.8	0.2	0.2	0.2	0.2	0.2
2good	0.8	0.7	0.7	0.2	0.2	0.2
3good	0.8	0.7	0.7	0.7	0.2	0.2
arith	0.8	0.7	0.575	0.45	0.325	0.2
geom	0.8	0.7	0.512	0.374	0.274	0.2

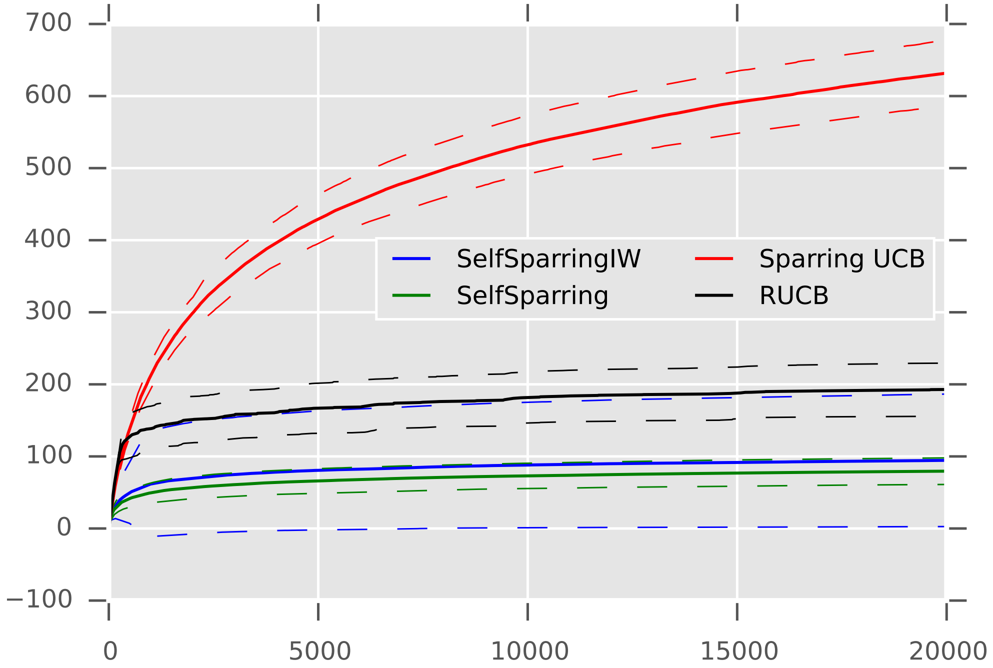


Figure 3.3: Expectations of cumulative regrets of top 4 algorithms. Dashed lines represent one standard deviation error curves.

Note that although these preference functions do not satisfy approximate linearity over their entire domains, they do for the utility samples (for arms A,B,C,D,E,F).

For each scenario, run each algorithm for 20000 iterations 48 times, and plot the averaged regret in Figure A.1. Set the ω parameter of SelfSparring-IW to be 2 in all trials. Some of these algorithms perform much better than others, so for clarity, also show a zoomed-in version of the linear/1good scenario, Figure 3.3, in which the expected regrets of just the top four algorithms (SelfSparring, SelfSparring-IW, RUCB and Sparring + UCB1) along with their one standard deviation curves are plotted.

I also investigate the effect of the scaling parameter ω on the empirical regret on SelfSparring-IW, and plot the results in Figure 3.4.

Results and Analysis. From Figure A.1, one can see that SelfSparring and SelfSparring-IW are consistently among the best algorithms, and outperform previous state-of-the-art Sparring UCB1 in every scenario. Only RUCB is competitive with SelfSparring-IW and SelfSparring; however, SelfSparring is approximately as

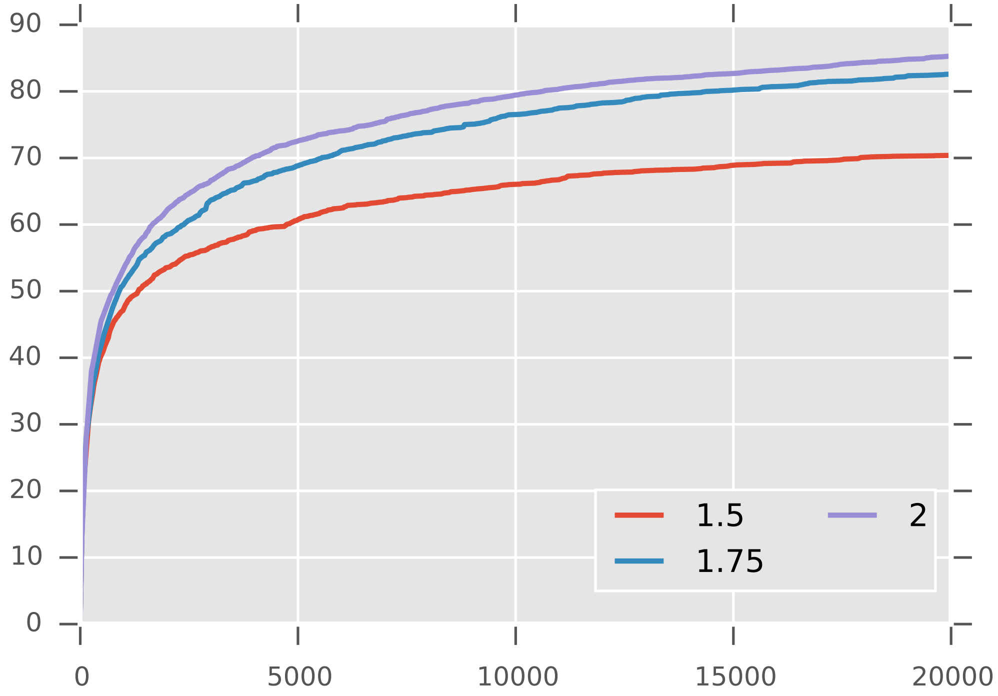


Figure 3.4: Expectations of cumulative regrets of SelfSparring-IW for $\omega = 1.5, 1.75, 2$.

good as or beats RUCB in every scenario. For example, from Figure 3.3, one can see the difference between SelfSparring/SelfSparring-IW and RUCB.

Figure 3.4 shows that the performance of SelfSparring-IW increases as ω decreases. In fact, as ω approaches 1, SelfSparring-IW becomes SelfSparring with minimal recalibration and importance weighting at every iteration. Furthermore, SelfSparring-IW suffers from high variance due to the importance-weighting scheme, as seen in Figure 3.3. However, observe that SelfSparring has variance comparable to that of Sparring UCB and RUCB, so due to its lower regret mean and variance, SelfSparring is preferred in practice.

3.6 Multi-dueling Bandits

We also extend the original dueling bandits problem by simultaneously dueling multiple arms as well as modeling dependencies between arms using a kernel. Explicitly formalizing these real-world characteristics provides an opportunity to develop principled algorithms that are much more efficient than algorithms designed for the original setting. For instance, most dueling bandits algorithms suffer regret that scales linearly with the number of arms, which is not practical when the number of arms is very large or infinite.

For this setting, we propose the SelfSparring framework, based on the dueling case SelfSparring algorithm from above, which algorithmically reduces the multi-dueling bandits problem into a conventional muilti-armed bandit problem that can be solved using a stochastic bandit algorithm such as Thompson Sampling (Chapelle and Li, 2011; Russo and Van Roy, 2014). Our approach can naturally incorporate dependencies using a Gaussian process prior with an appropriate kernel.

While there have been some prior work on multi-dueling (Brost et al., 2016) and learning from pairwise preferences over kernels (Gonzalez et al., 2016), to the best of our knowledge, our approach is the first to address to both in a unified framework. We are also the first to provide a regret analysis of the multi-dueling setting. We further demonstrate the effectiveness of our approach over conventional dueling bandits approaches in a wide range of simulation experiments.

We now formalize the multi-dueling bandits problem. We inherit all notation from original dueling bandits setting (Section 2.4). The key difference is that the algorithm now selects a (multi-)set S_t of arms at each iteration t, and observes outcomes of duels between some pairs of arms in S_t. For example, in information retrieval this can be implemented via multi-leaving (Schuth et al., 2014) the ranked lists of the subset, S_t, of rankers and then inferring the relative quality of the lists (and the corresponding rankers) from user feedback.

In general, we assume the number of arms being dueled at each iteration is some fixed constant $m = |S_t|$. When $m = 2$, the problem reduces to the original dueling bandits setting. Extending the regret formulation from the original setting (3.1), we can write the regret as:

$$R_T = \sum_{t=1}^{T} \sum_{b \in S_t} \phi(b_1, b). \tag{3.5}$$

The goal then is to select subsets of arms S_t so that the cumulative regret (3.5) is minimized. Intuitively, all arms have to be selected a small number of times in order to be explored, but the goal of the algorithm is to minimize the number of times when suboptimal arms are selected. When the algorithm has converged to the best arm b_1, then it can simply choose S_t to only contain b_1, thus incurring no additional regret.

Our setting differs from Brost et al. (2016) in two ways. First, we play a fixed, rather than variable, number of arms at each iteration. Furthermore, we focus on total regret, rather than the instantaneous average regret in a single iteration; in many applications (e.g., Sui and J. Burdick (2014b)), playing each arm incurs its own regret .

Feedback Mechanisms. Simultaneously dueling multiple arms opens up multiple options for collecting feedback. For example, in some applications it may be viable to collect all pairwise feedback for all chosen arms S_t. In other applications, it is more realistic to only observe the "winner" of S_t, in which we observe feedback that one $b \in S_t$ wins against all other arms in S_t, but nothing about pairwise preferences between the other arms.

3.7 SelfSparring Algorithms

We start with a high-level description of our general framework, called SelfSparring, which is inspired by the Sparring algorithm from Ailon, Z. Karnin, and Joachims (2014). The high-level strategy is to reduce the multi-dueling bandits problem to a multi-armed bandit (MAB) problem that can be solved using a MAB algorithm, and ideally lift existing MAB guarantees to the multi-dueling setting.

Algorithm 5 describes the SelfSparring approach. SelfSparring uses a stochastic MAB algorithm such as Thompson sampling as a subroutine to independently sample the set of m arms, S_t to duel. The distribution of S_t is generally not degenerate (e.g., all the same arm) unless the algorithm has converged. In contrast, the Sparring algorithm uses m MAB algorithms to control the choice of the each arm, which essentially reduces the conventional dueling bandits problem to two multi-armed bandit problems "sparring" against each other.

SelfSparring takes as input S the total set of arms, m the number of arms to be dueled at each iteration, and η the learning rate for posterior updates. S can be a finite set of K arms for independent setting, or a continuous action space of arms for kernelized setting. A prior distribution D_0 is used to initialize the sampling process over S. In the t-th iteration, SelfSparring selects m arms by sampling over the distribution D_{t-1} as shown in line 5 of Algorithm 5. The preference feedback can be any type of comparisons ranging from full comparison over the m arms (a full matrix for R, aka 'all pairs") to single comparison of one pair (just two valid entries in R). The posterior distribution over arms D_t then gets updated by R and the prior D_{t-1}.

We specialize SelfSparring in two ways. The first, IndSelfSparring (Algorithm 6),

Algorithm 5 SelfSparring

```
input  arms $1, \ldots, K$ in space $S$, $m$ the number of arms drawn at each iteration, $\eta$ the learning rate
Set prior $D_0$ over $S$
for $t = 1, 2, \ldots$ do
    for $j = 1, \ldots, m$ do
        select arm $i_j(t)$ using $D_{t-1}$
    end for
    Play $m$ arms $\{i_j(t)\}_j$ and observe $m \times m$ pairwise feedback matrix $R = \{r_{ij} \in \{0, 1, \emptyset\}\}_{m\times m}$
    update $D_{t-1}$ using $R$ to obtain $D_t$
end for
```

is the independent-armed version of SelfSparring. The second, KernelSelfSparring (Algorithm 7), uses Gaussian processes to make predictions about preference function f based on noisy evaluations over comparisons. We emphasize here that SelfSparring is very modular approach, and is thus easy to implement and extend.

Independent Arms Case

IndSelfSparring (Algorithm 6) instantiates SelfSparring using Beta-Bernoulli Thompson sampling.

The posterior Beta distributions D_t over the arms are updated by the preference feedback within the iteration and the prior Beta distributions D_{t-1}.

We present a no-regret guarantee of IndSelfSparring in Theorem 2 below. We now provide a high-level outline of the main components leading to the result. Detail proofs are deferred to the supplementary material.

Our first step is to prove that IndSelfSparring is asymptotically consistent, i.e., it is guaranteed (with high probability) to converge to the best bandit. In order to guarantee consistency, we first show that all arms are sampled infinitely often in the limit.

Lemma 4 *For the K-armed stochastic MAB problem, Thompson Sampling has expected regret:* $\mathbb{E}[R_T^{MAB}] = O\left(\frac{K}{\Delta}\ln T\right)$*, where* Δ *is the difference between expected rewards of the best two arms.*

Lemma 5 *Running IndSelfSparring with infinite time horizon will sample each arm infinitely often.*

Algorithm 6 IndSelfSparring

```
input  $m$ the number of arms drawn at each iteration, $\eta$ the learning rate
For each arm $i = 1, 2, \cdots, K$, set $S_i = 0$, $F_i = 0$.
for $t = 1, 2, \ldots$ do
    for $j = 1, \ldots, m$ do
        For each arm $i = 1, 2, \cdots, K$, sample $\theta_i$ from $Beta(S_i + 1, F_i + 1)$
        Select $i_j(t) := \mathrm{argmax}_i\, \theta_i(t)$
    end for
    Play $m$ arms $\{i_j(t)\}_j$, observe pairwise feedback matrix $R = \{r_{jk} \in \{0, 1, \emptyset\}\}_{m \times m}$
    for $j, k = 1, \ldots, m$ do
        if $r_{jk} \neq \emptyset$ then
            $S_j \leftarrow S_j + \eta \cdot r_{jk}$, $F_j \leftarrow F_j + \eta(1 - r_{jk})$
        end if
    end for
end for
```

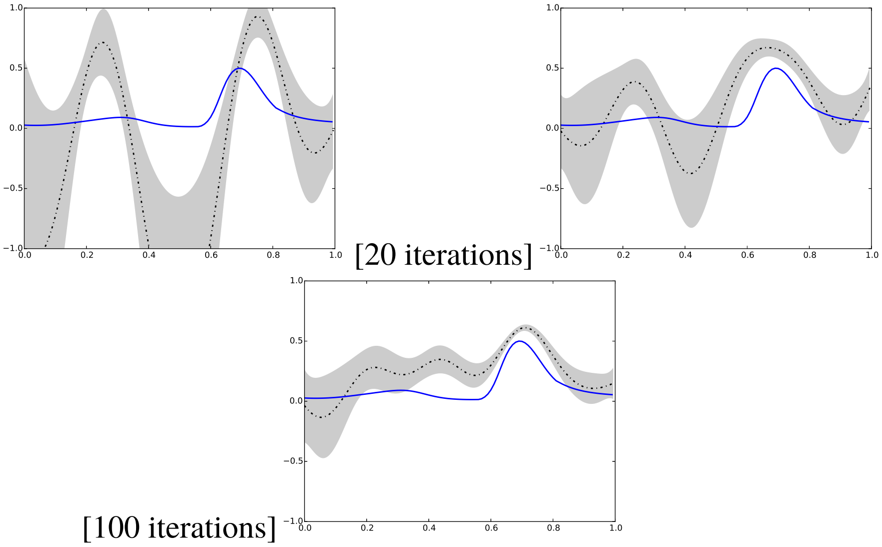
In other words, Thompson sampling style algorithms do not eliminate any arms. Lemma 5 also guarantees concentration of any statistical estimates for each arm as $t \rightarrow \infty$. We next show that the sampling of IndSelfSparring will concentrate around the optimal arm.

Theorem 1 *Under Approximate Linearity, IndSelfSparring converges to the optimal arm b_1 as running time $t \rightarrow \infty$:* $\lim_{t \rightarrow \infty} \mathbb{P}(b_t = b_1) = 1$.

Theorem 1 implies that IndSelfSparring is asymptotically no-regret. As $t \rightarrow \infty$, the Beta distribution for each arm i is converging to $P(b_i \succ b_1)$, which implies converging to only choosing the optimal arm.

Most existing dueling bandits algorithm chooses one arm as a “reference” arm and the other arm as a competing arm for exploration/exploitation (in the $m = 2$ setting). If the distribution over reference arms never changes, then the competing arm is playing against a fixed “environment”, i.e., it is a standard MAB problem. For general m, we can analogously consider choosing only one arm against a fixed distribution over all the other arms. Using Thompson sampling, the following lemma holds.

Lemma 6 *Under Approximate Linearity, selecting only one arm via Thompson sampling against a fixed distribution over the remaining arms leads to optimal regret w.r.t. choosing that arm.*



[5 iterations] [20 iterations]

[100 iterations]

Figure 3.5: Evolution of a GP preference function in KernelSelfSparring; dashed lines correspond to the mean and shaded areas to ±2 standard deviations. The underlying utility function was sampled randomly from a GP with a squared exponential kernel with lengthscale parameter 0.2, and the resulting preference function is shown in blue. The GP finds the best arm with high confidence.

Lemma 6 and Theorem 1 motivate the idea of analyzing the regret of each individual arm against near-fixed (i.e., converging) environments.

Theorem 2 *Under Approximate Linearity, IndSelfSparring converges to the optimal arm with asymptotically optimal no-regret rate of* $O(K \ln(T)/\Delta)$.

Theorem 2 shows an no-regret guarantee for IndSelfSparring that asymptotically matches the optimal rate of $O(K \ln(T)/\Delta)$ up to constant factors. In other words, once $t > C$ for some problem-dependent constant C, the regret of IndSelfSparring matches information-theoretic bounds up to constant factors (see Yue et al. (2012) for lower bound analysis).[4] The proof technique follows two major steps: (1) prove the convergence of IndSelfSparring as shown in Theorem 1; and (2) bound the expected total regret for sufficiently large T.

Dependent Arms Case

We use Gaussian processes (see Section 2.5) to model dependencies among arms. Applying Gaussian processes is not straightforward, since the underlying utility

[4] A finite-time guarantee requires more a refined analysis of C, and is an interesting direction for future work.

Algorithm 7 KernelSelfSparring

```
input  Input space $S$, GP prior $(\mu_0, \sigma_0)$, $m$ the number of arms drawn at each iteration
for $t = 1, 2, \ldots$ do
    for $j = 1, \ldots, m$ do
        Sample $f_j$ from $(\mu_{t-1}, \sigma_{t-1})$
        Select $i_j(t) := \text{argmax}_x f_j(x)$
    end for
    Play $m$ arms $\{i_j(t)\}_j$, observe pairwise feedback matrix $R = \{r_{jk} \in \{0, 1, \emptyset\}\}_{m \times m}$
    for $j, k = 1, \ldots, m$ do
        if $r_{jk} \neq \emptyset$ then
            apply Bayesian update using $(i_j(t), r_{jk})$ to obtain $(\mu_t, \sigma_t)$
        end if
    end for
end for
```

function is not directly observable or does not exist. We instead use Gaussian processes to model a specific the preference function. In Gaussian process notation, the preference function $f(b)$ represents the preference of choosing b over the perfect "environment" of competing arms. Like in the independent arms case (Section 3.7), the perfect environment corresponds to having all the remaining arms be deterministically selected as the best arm b_1, yielding $f(b) = P(b \succ b_1)$. We model $f(b)$ as a sample from a Gaussian process $GP(\mu(b), k(b, b'))$. Note that this setup is analogous to the independent arms case, which uses a Beta prior to estimate the probability of each arm defeating the environment (and converges to competing against the best environment).

Algorithm 7 describes KernelSelfSparring, which instantiates SelfSparring using a Gaussian process Thompson sampling algorithm. The input space S can be continuous. At each iteration t, m arms are sampled using the Gaussian process prior D_{t-1}. The posterior D_t is then updated by the responses R and the prior.

Figure 3.5 illustrates the optimization process in a one-dimensional example. The underlying preference function against the best environment is shown in blue. Dashed lines are the mean function of GP. Shaded areas are ±2 standard deviations regions (high confidence regions). Figures 3.5(a)(b)(c) represent running KernelSelfSparring algorithm at 5, 20, and 100 iterations. The GP model can be observed to be converging to the preference function against the best environment.

We conjecture that it is possible to prove no-regret guarantees that scale w.r.t. the dimensionality of the kernel. However, there does not yet exist suitable regret

Name	Distribution of Utilities of arms
1good	1 arm with utility 0.8, 15 arms with utility 0.2
arith	1 arm with utility 0.8, 15 arms forming an arithmetic sequence between 0.7 and 0.2

Table 3.2: 16-arm synthetic datasets used for experiments.

analyses for Gaussian Process Thompson Sampling in the kernelized MAB setting to leverage.

3.8 SelfSparring Experiments

Simulation Settings & Datasets

Synthetic Functions. We evaluated on a range of 16-arm synthetic settings derived from the utility-based dueling bandits setting of Ailon, Z. Karnin, and Joachims (2014). For the multi-dueling setting, we used the following preference functions:

$$\begin{array}{ll} \text{linear:} & \phi(x,y) - 1/2 = (1 + x - y)/2 \\ \text{logit:} & \phi(x,y) - 1/2 = (1 + \exp(y - x))^{-1} \end{array}$$

and the utility functions shown in Table 3.2 (generalized from those in Ailon, Z. Karnin, and Joachims (2014)). Note that although these preference functions do not satisfy approximate linearity over their entire domains, they do for the utility samples (over the a finite subset of arms).

MSLR Dataset. Following the evaluation setup of Brost et al. (2016), we also used the Microsoft Learning to Rank (MSLR) WEB30k dataset, which consists of over 3 million query-document pairs labeled with relevance scores (T.-Y. Liu et al., 2007). Each pair is scored along 136 features, which can be treated as rankers (arms). For any subset of arms, we can estimate a preference matrix using the expected probability over the entire dataset of one arm beating another using top-10 interleaving and a perfect-click model. We simulate user feedback by using team-draft multileaving (Schuth et al., 2014).

Vanilla Dueling Bandits Experiments

We first compare against the vanilla dueling bandits setting of dueling a single pair of arms at a time. These experiments are included as a sanity check to confirm that SelfSparring (with $m = 2$) is a competitive algorithm in the original dueling bandits setting, and are not the main focus of our empirical analysis.

We empirically evaluate against a range of conventional dueling bandit algorithms, including:

- **Interleaved Filter (IF)** (Yue et al., 2012)
- **Beat the Mean (BTM)** (Yue and Joachims, 2011)
- **RUCB** (Zoghi, Whiteson, Remi Munos, et al., 2014)
- **MergeRUCB** (Zoghi, Whiteson, and Rijke, 2015)
- **Sparring + UCB1** (Ailon, Z. Karnin, and Joachims, 2014)
- **Sparring + EXP3** (Dudik et al., 2015)
- **RMED1** (Komiyama et al., 2015)
- **Double Thompson Sampling** (**wu2016doublets**)

For Double Thompson Sampling and IndSelfSparring, we set the learning rates to be 2.5 and 3.5 as optimized over a separate dataset of uniformly sampled utility functions. We use $\alpha = 0.51$ for RUCB/MergeRUCB, $\gamma = 1$ for BTM, and $f(K) = 0.3K^{1.01}$ for RMED1.

Results. For each scenario, we run each algorithm 100 times for 20000 iterations. For brevity, we show in Figure 3.6 the average regret of one synthetic simulation along with shaded one standard-deviation areas. We observe that SelfSparring is competitive with the best performing methods in the original dueling bandits setting. More complete experiments that replicate Ailon, Z. Karnin, and Joachims (2014) are provided in the supplementary material, and demonstrate the consistency of this result.

Double Thompson Sampling (DTS) is the best performing approach in Figure 3.6, which is a fairly consistent result in the extended results in the supplementary material. However, given their high variances they are essentially comparable w.r.t. all other algorithms. Furthermore, IndSelfSparring has the advantage of being easily extensible to the more realistic multi-dueling and kernelized settings, which is not true of DTS.

Multi-Dueling Bandits Experiments

Next evaluate the multi-dueling setting with independent arms. Compare against the main existing approaches that are applicable to the multi-dueling setting, including the MDB algorithm (Brost et al., 2016), and the multi-dueling extension of Sparring, which we refer to as MultiSparring (Ailon, Z. Karnin, and Joachims, 2014). Following Brost et al., 2016, we use $\alpha = 0.5$ and $\beta = 1.5$ for the MDB algorithm. For IndSelfSparring, we set learning rate to be the default 1. Note that the vast majority dueling bandits algorithms are not easily applicable to the multi-dueling

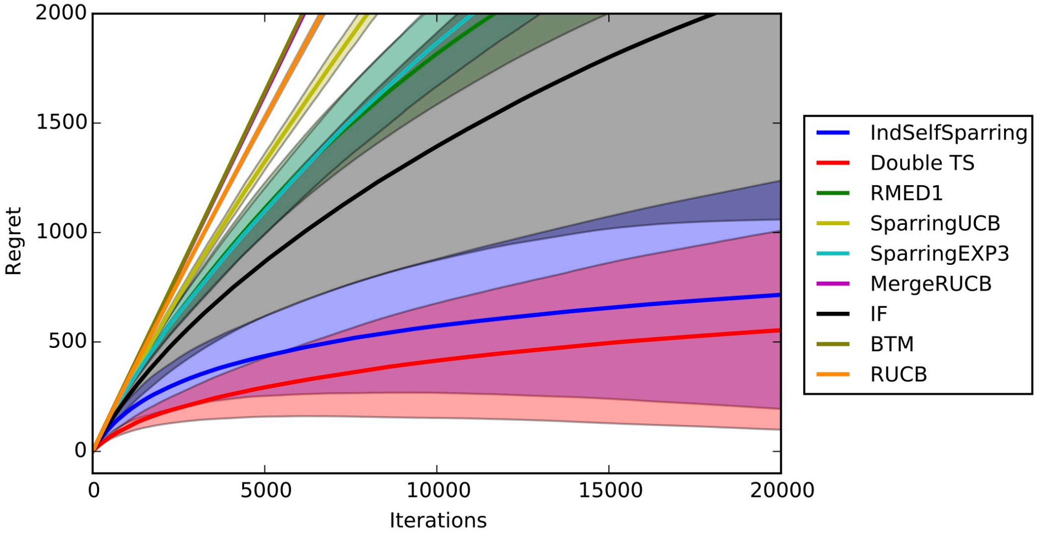


Figure 3.6: Vanilla dueling bandits setting. Average regret for top nine algorithms on logit/arith. Shaded regions correspond to one standard deviation.

setting. For instance, RUCB-style algorithms treat the two arms asymmetrically, which is not easily generalized to multi-dueling.

Results on Synthetic Experiments. Test $m = 4$ on the linear 1good and arith datasets in Figure 3.7 and Figure 3.8, respectively. It is observed that IndSelfSparring significantly outperforms competing approaches.

Results on MSLR Dataset. Following the simulation setting of Brost et al., 2016 on the MSLR dataset (see Section 3.8), SelfSparring is compared against the MDB algorithm over the same collection of 50 randomly sampled 16-arm subsets. Ensuring that each 16-arm subset had a Condorcet winner; in general it is likely for any random subset of arms in the MSLR dataset to have a Condorcet winner (Zoghi, Z. S. Karnin, et al., 2015). Figure 3.9 shows the results, where one can see that IndSelfSparring enjoys significantly better performance.

Kernelized (Multi-)Dueling Experiments

This subsection evaluates the kernelized setting for both the 2-dueling and the multi-dueling case. KernelSelfSparring is evaluated against BOPPER (Gonzalez et al., 2016) and Sparring (Ailon, Z. Karnin, and Joachims, 2014) with GP-UCB (Srinivas et al., 2010a). BOPPER is a Bayesian optimization method can be applied to kernelized 2-dueling setting (but not multi-dueling). Sparring with GP-UCB, which refer to as GP-Sparring, is essentially a variant of our KernelSelfSparring approach

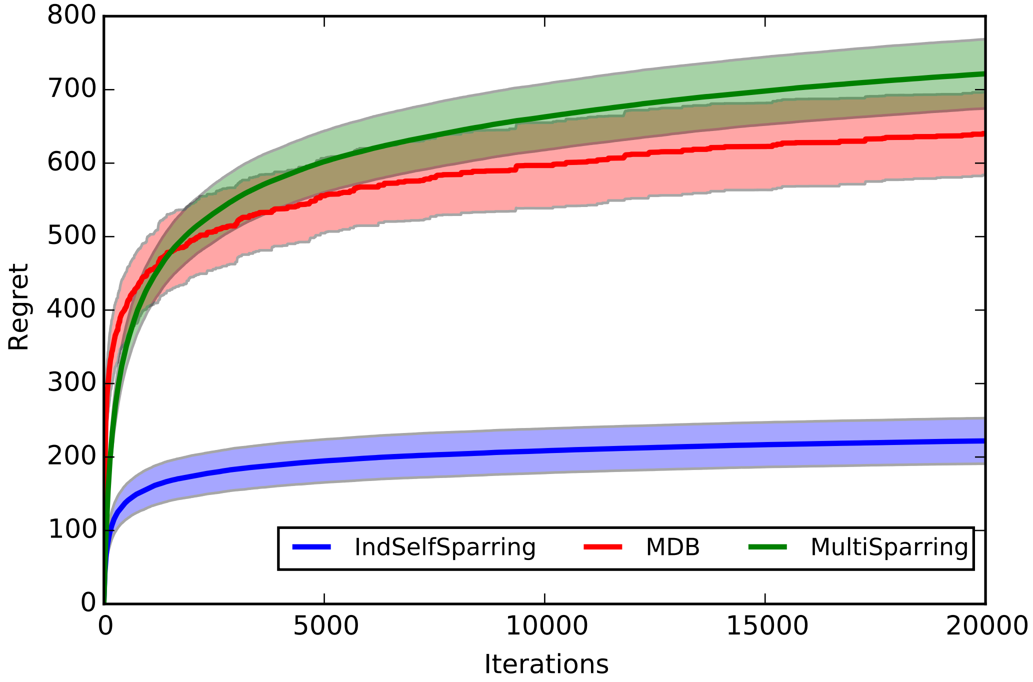


Figure 3.7: Multi-dueling regret for linear/1good setting

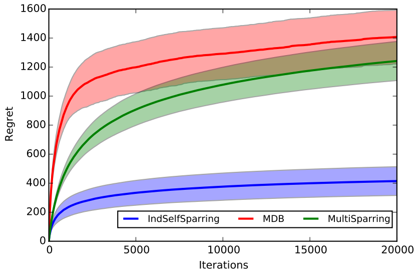


Figure 3.8: Multi-dueling regret for linear/arith setting

but maintains a m GP-UCB bandit algorithms (one controlling each choice of arm to be dueled), rather than just a single one.

KernelSelfSparring and GP-Sparring use GPs that model the preference function, i.e. are one-sided, whereas BOPPER uses a GP to model the entire preference matrix. Following Srinivas et al. (2010a), a squared exponential kernel is used with lengthscale parameter 0.2 for both GP-Sparring and KernelSelfSparring, and use a squared exponential kernel with parameter 1 for BOPPER. Initialize all GPs with a zero-mean prior, and use sampling noise variance $\sigma^2 = 0.025$. For GP-Sparring, use the scaled-down version of β_t as suggested by Srinivas et al. (2010a).

Use the Forrester and Six-Hump Camel functions as utility functions on $[0, 1]$ and

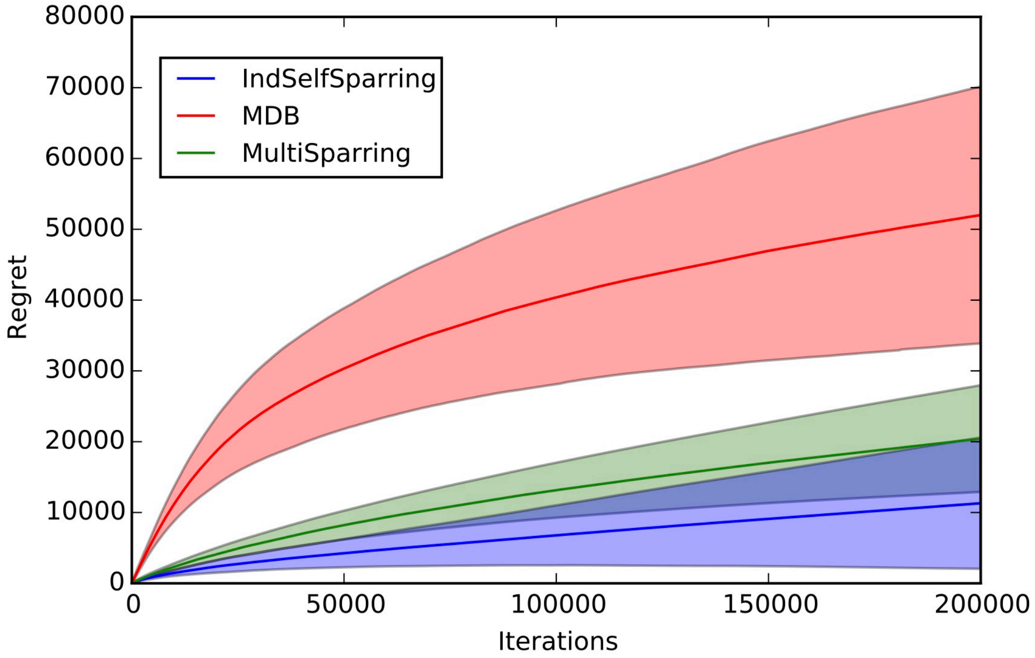


Figure 3.9: Multi-dueling regret for MSLR-30K experiments

$[0, 1]^2$, respectively, as in Gonzalez et al. (2016). Similarly, use the same uniform discretizations of 30 and 64 points for the Forrester and Six-Hump Camel settings respectively, and use the logit link function to generate preferences.

Since the BOPPER algorithm is computationally expensive, only including it in the Forrester setting, and running each algorithm 20 times for 100 iterations. In the Six-Hump Camel setting, we run KernelSelfSparring and GP-Sparring for 500 iterations 100 times each. Results are presented in Figures 3.11 and 3.12, where we observe much better performance from KernelSelfSparring against both BOPPER and GP-Sparring.

In the kernelized multi-dueling setting, SelfSparring is compared against GP-Sparring. Running each algorithm for 100 iterations 50 times on the Forrester and Six-Hump Camel functions, and plotting their regrets in Figures 3.13 and 3.14 respectively. Use $m = 4$ for both algorithms, and the same discretization as in the standard dueling case. One can observe significant performance gains of our KernelSelfSparring approach.

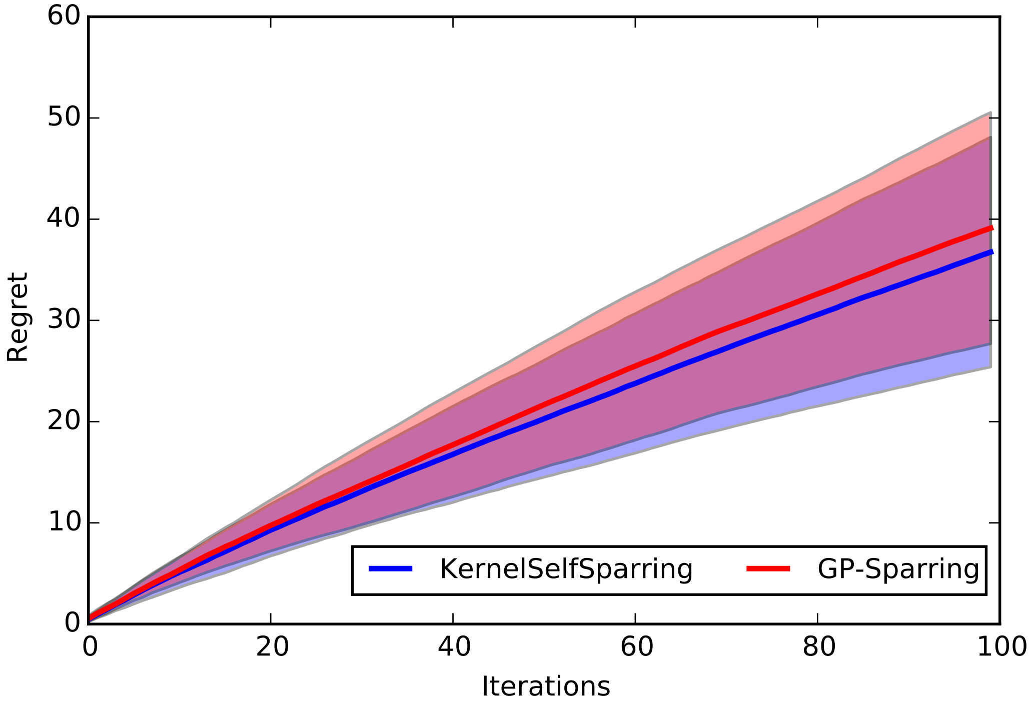


Figure 3.10: 2-dueling regret for kernelized setting with synthetic preferences

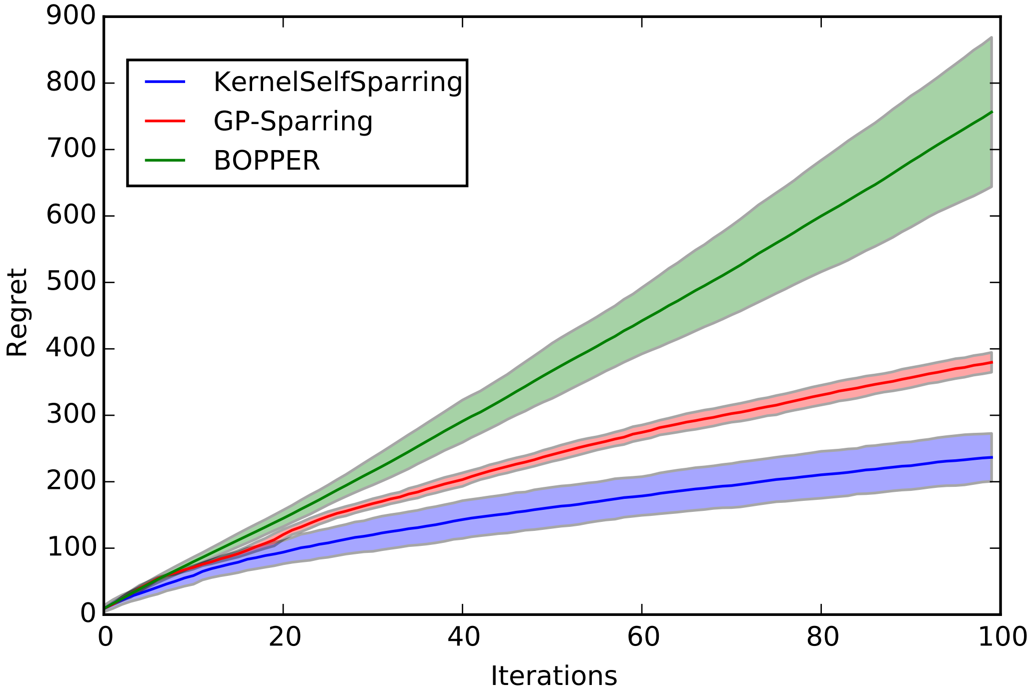


Figure 3.11: 2-dueling regret for kernelized setting with Forrester objective function

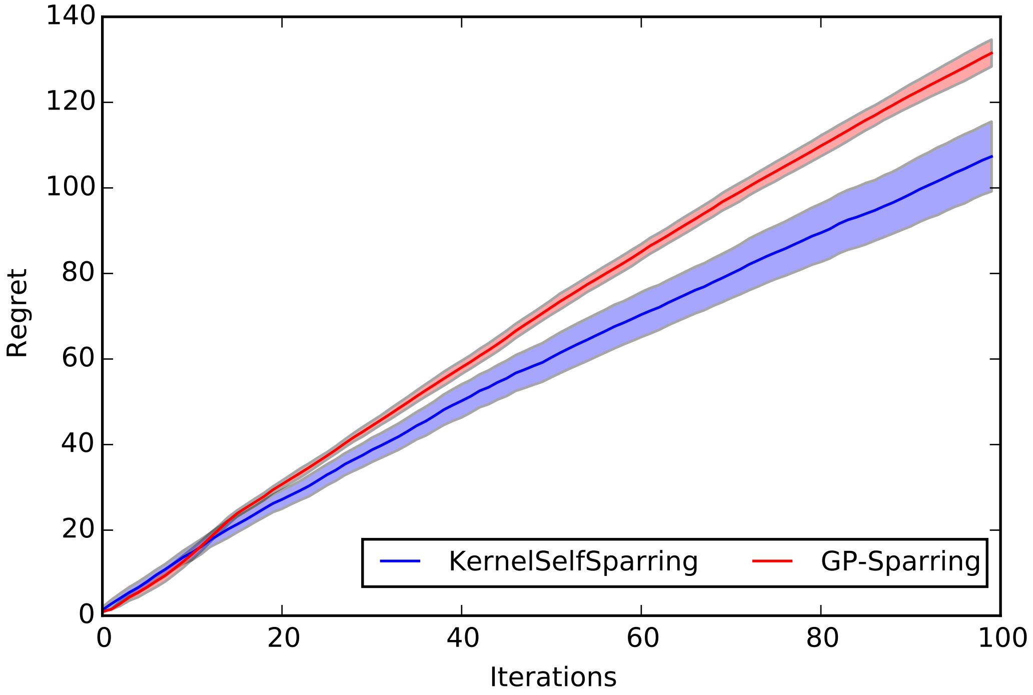


Figure 3.12: 2-dueling regret for kernelized setting with Six-Hump Camel objective function

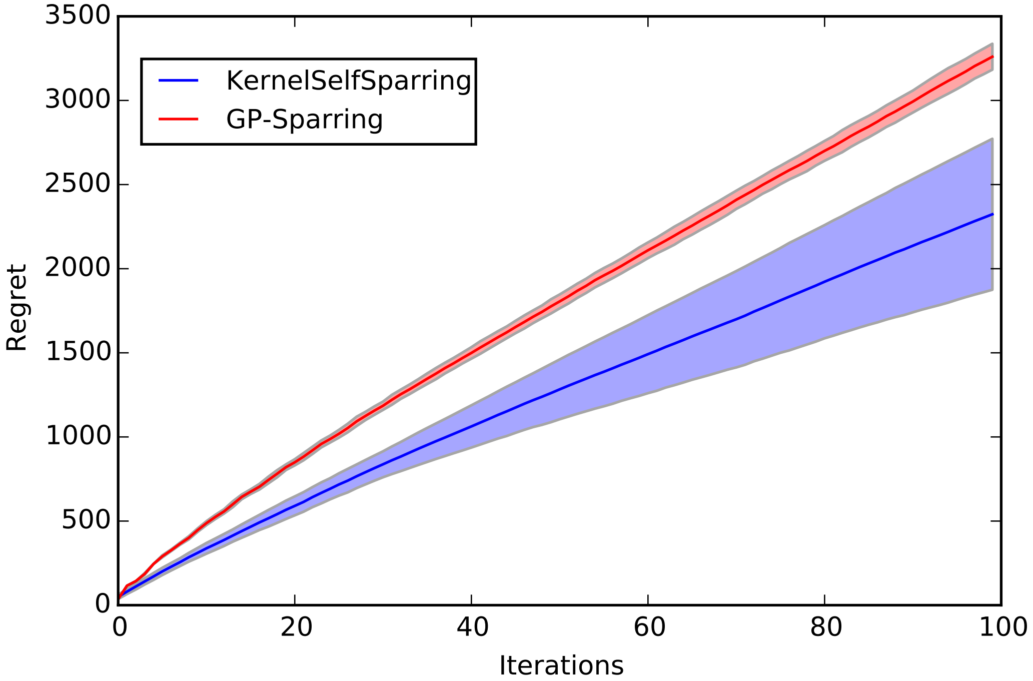


Figure 3.13: Multi-dueling regret for kernelized setting with Forrester objective function

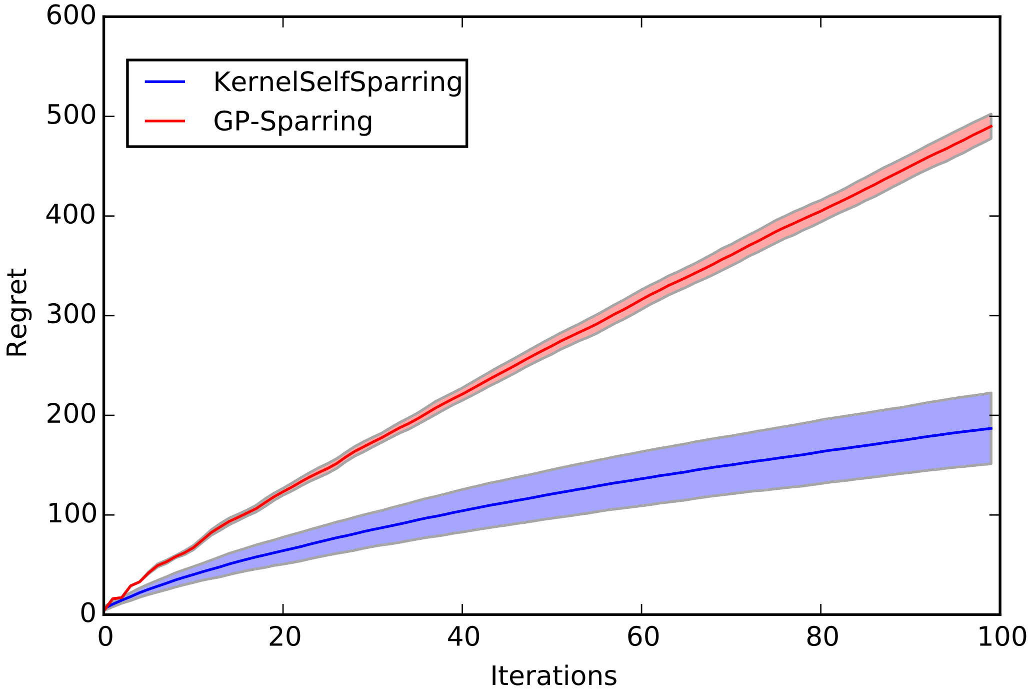


Figure 3.14: Multi-dueling regret for kernelized setting with Six-Hump Camel objective function

Chapter 4

THEORETICAL CONTRIBUTIONS: RANK-COMPARISON ALGORITHM FOR DUELING BANDITS

The original dueling bandits problem described in Chapter 3 suffers from a theoretical lower bound $\frac{K}{\delta} \log T$ for cumulative regret. To achieve even lower regret (which translates to better performance), more assumptions or finer problem formulations are necessary. This chapter studies the Multi-armed bandit problem with feedback given as a stochastic rank list instead of quantified reward values. An algorithm, RankComparison, is proposed for this new problem, with theoretical guarantees on the optimality of total regret.

4.1 Introduction and Motivation

Figure 4.1 shows the clinical treatment procedure for *stand-training*. During a treatment/optimization session, a new stimulus is recommended by our algorithm. The patient then attempts to stand using the given stimulus, and the observing clinicians then rank the patient's resulting performance. Using this noisy ranking

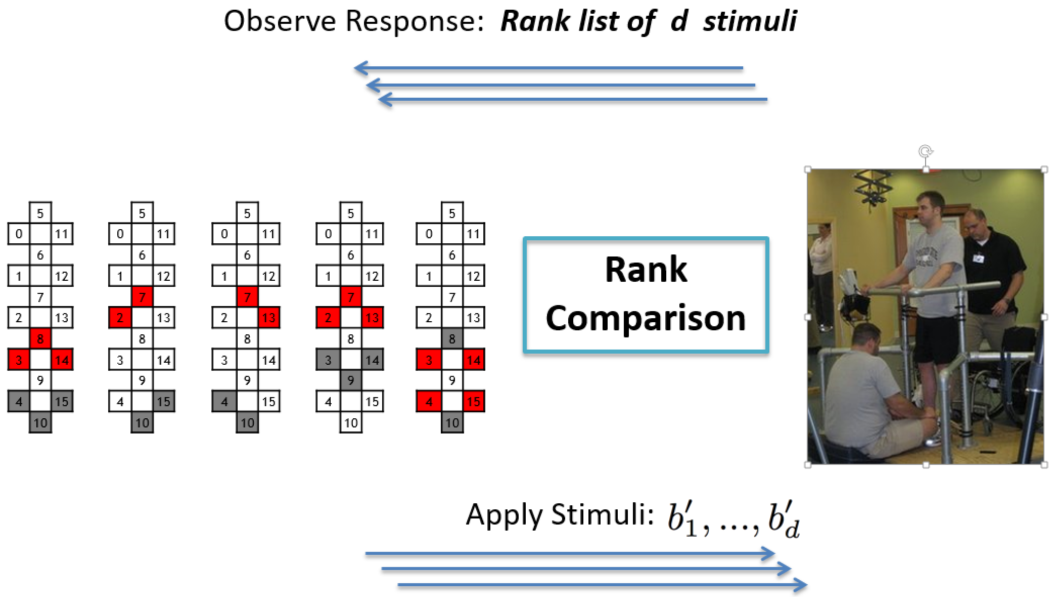

Observe Response: ***Rank list of d stimuli***
Rank Comparison
Apply Stimuli: $b'_1, \ldots, b'_d$

Figure 4.1: Clinical Treatment of Spinal Cord Injury

as feedback, the algorithm continues to explore for the optimal stimulus while also exploiting currently good ones. The algorithm must spend significant time dwelling on good performing stimuli in order to provide the patient with a good therapeutic experience. Since clinical training has a fixed time horizon, we must also maximize total performance during the limited period within which we can search for the optimal solution.

This chapter develops an algorithm to recommend optimal stimuli based on the general setting of multi-armed bandit problem. The classical bandit problem trades off between exploration and multi-armed bandit problem exploitation among a number of different arms, each having a quantifiable, but stochastic, reward with initially unknown distribution. The goal of a bandit algorithm is to maximize the total reward.

However, for our clinical problem, the patient's motor response to stimulation is hard to quantify. Neither video motion capture nor electromyograhic (EMG) recordings of muscle activity can yet provide a consistent and satisfactory measure of motor skill under stimulation. A good standing performance might map to numerous combinations of muscle activities, and it is not a stationary process. While the patient's performance under a specific stimulus is hard to quantify, it can be compared to others. In the clinical setting, we can obtain the ranking of a *group* of stimuli which are tested within the short time period of one training session. The *dueling bandit problem* (Yue et al., 2009) formalizes online learning problems with preference feedback instead of absolute rewards, and hence it can be used for problems with unquantifiable reward. The algorithm we propose in this chapter is a variant of the dueling bandit problem which is dictated by the clinical demands of our application.

At the start of the optimization process, we have little information about the best stimulus for the patient, but we have often have a pool of possibly useful stimuli. Like Sparring, this approach is based on the idea of successively removing suboptimal arms (Even-Dar, Mannor, and Mansour, 2002) while keeping the optimal one(s) in the sample space. By setting proper confidence intervals, we can reach the optimal reward within the time horizon.

4.2 Problem Setup

The classical dueling bandit problem receives feedback in the form of a comparison between a pair of bandits in each test. When the size of the decision set, K, is large, it is unavoidable to carry out a very large number of tests before the algorithm converges to its optimal solution. In some applications like our clinical example, each test is expensive and time consuming. The number of tests - time horizon of an algorithm - is often predetermined by clinical conditions. It is infeasible to apply the dueling bandit algorithm directly.

However, the training and optimization procedure allows for patients to not only compare successive stimulations, but to also rank the performances for a modest-sized group of stimulations (the number which can be tested in one clinical session before the patient fatigues). Thus, feedback consists of a ranked list of at most d ($d < K$) chosen arms. More precisely, the feedback for each test consists of a combined scoring of 4 different standing criteria by the observing clinicians, and the combined score is used to rank the tests within one session. As shown below, this feature helps us to reduce the total number of tests significantly, while also dovetailing well with current clinical practice.

The procedure can be described as follows. There are K arms $\{b_1, \cdots, b_K\}$, and a total number of T tests to be performed. Each test physically corresponds to a ~90-second stimulation period with a specific stimulus (arm) chosen from the K arms. T is determined before we run the algorithm, and is generally assumed to be an integer multiple of d: $T = d * G$, where G is the number of ranking sessions, with each session producing a noisy ranked list of d arms.

This approach follows the the original notation of the dueling bandit problem (Yue et al., 2009). For two arms b_i and b_j, where $i, j \in \{1, \cdots, K\}$, write the comparison factor as:

$$\epsilon(b_i, b_j) = P(b_i \succ b_j) - 1/2$$

where $P(b_i \succ b_j)$ is the probability that b_i dominates b_j and $\epsilon(b_i, b_j) \in [-1/2, 1/2]$ represents the priority between b_i and b_j. We define $b_i \succ b_j \Leftrightarrow \epsilon(b_i, b_j) > 0$. Use the notation $\epsilon_{i,j} \equiv \epsilon(b_i, b_j)$ for convenience. Note that $\epsilon(b_i, b_j) = -\epsilon(b_j, b_i)$ and $epsilon(b_i, b_i) = 0$. Assume the distribution of reward for each arm is stationary so that all comparison factors converge in [-1/2,1/2]. This setup also assumes without loss of generality that the bandits are indexed in preferential order $b_1 \succ b_2 \succ \cdots \succ b_K$ so that there is one preferred arm.

The total reward is defined in terms of regret as in the classical bandit problem setting. In the online setting, let $b_{(t)}$ be the arm chosen at test t. Define total regret as follows:

$$R_T = \sum_{t=1}^{T} \epsilon(b_1, b_{(t)}).$$

Algorithm 8 Rank-Comparisons

```
Input: $\{b_1, ..., b_K\}, d, G$   // $Total\ tests\ T = d \cdot G$
Input: $c_\delta(n) = \sqrt{(1/n)log(1/\delta)}$
Run: [Parameters-Initialization]
Run: [Active-Elimination]
return $b^*$   // $Optimal\ arm$
```

Algorithm 9 Parameters-Initialization

```
Input: $\{b_1, ..., b_K\}, d, G$
Input: $c_\delta(n) = \sqrt{(1/n)log(1/\delta)}$
$W_1 \longleftarrow \{b_1, ..., b_K\}$   // $set\ of\ active\ arms$
$\ell \longleftarrow 1$   // $rounds$
$\forall b \in W_\ell,\ n_b \longleftarrow 0$   // $comparisons$
$\forall b \in W_\ell,\ w_b \longleftarrow 0$   // $priorities$
$\forall b \in W_\ell,\ \hat{P}_b \equiv w_b/n_b$, or 1/2 if $n_b = 0$
$n^* \equiv min_{b \in W_\ell} n_b$
$c^* \equiv c_\delta(n^*)$, or 1 if $n^* = 0$   // $confidence\ radius$
$g \longleftarrow 0$   // $total\ number\ of\ ranks$
$T \longleftarrow d \cdot G$
return all new parameters
```

The total regret $R_T = 0$ if we constantly choose $b_{(t)} = b_1$ during the experiment. $R_T = \Theta(T)$ is linear $w.r.t.$ T if the agent constantly chooses $b_{(t)} \in \{b_1, \cdots, b_K\}$.

This setup also inherit two important properties of the comparison factors from the original dueling bandit problem:

Strong Stochastic Transitivity. For any triplet of arms $b_i \succ b_j \succ b_k$, we assume $\epsilon_{i,k} \geq \max\{\epsilon_{i,j}, \epsilon_{j,k}\}$.

Stochastic Triangle Inequality. For any triplet of arms $b_i \succ b_j \succ b_k$, we assume $\epsilon_{i,k} \leq \epsilon_{i,j} + \epsilon_{j,k}$. This can be viewed as a diminishing returns property.

An optimal method is proposed for our problem which has a finite-time regret bound of order $O(\frac{K}{d} logT)$, where T is the time horizon.

4.3 Algorithm

Our *Rank-Comparison* algorithm (Algorithm 1), which is a modified version of "Beat-the-Mean" Yue and Joachims (2011), is based on the idea of successively removing suboptimal arms while keeping the optimal one(s) in the sample space. The inputs to *Rank-Comparison* are the K arms, the largest group size d, and total number of groups G: $T = d \cdot G$.

Algorithm 10 Active-Elimination

```
Input: $\{b_1, ..., b_K\}$, $d$, $G$
Input: parameters generated in [Parameters-Initialization]
while $|W_\ell| > 1$ and $g \leq G$ do
    if $|W_\ell| \geq d$ then
        select $b'_1, ..., b'_d \in W_\ell$ at random with no repeats
    else
        $r \leftarrow d\%|W_\ell|$
        $p \leftarrow (d - r)/|W_\ell|$
        select $b'_1, ..., b'_r \in W_\ell$ at random with no repeats. In addition, select each
        arm in $W_\ell$ $p$ times
    end if
    test selected arms and get rank of the selection
    for all commutable pairs $(b'_i, b'_j)$ in the selection do
        if $b'_i \succ b'_j$, $w_{b'_i} \leftarrow w_{b'_i} + 1$
        $n_{b'_i} \leftarrow n_{b'_i} + 1$
        if $\min_{b' \in W_\ell} \hat{P}_{b'} + c^* \leq \max_{b \in W_\ell} \hat{P}_b - c^*$ then
            $b' \leftarrow \arg\min_{b \in W_\ell} \hat{P}_b$
            $\forall b \in W_\ell$, delete comparisons with $b'$ from $w_b$, $n_b$
            $W_{\ell+1} \leftarrow W_\ell \backslash \{b'\}$    // update working set
            $\ell \leftarrow \ell + 1$    // new round
        end if
    end for
end while
return $b^* = \arg\max_{b \in W_\ell} \hat{P}_b$
```

Parameters-Initialization (Algorithm 2) defines the set of active arms W_ℓ, whose size shrinks as more tests are completed. For each arm b, let n_b be the total number of comparisons between b and other arms, and let w_b be the total number of wins against all other arms. Let $\hat{P}_b$ be the empirical average of $P(b \succ b')$ for all b' in W_ℓ, and let $\hat{P}_{b,n}$ be the value of $\hat{P}_b$ after n comparisons between arm b and any other arms. Set the confidence interval of $P(b \succ b')$ as:

$$\hat{C}_{b,n} = (\hat{P}_{b,n} - c_\delta(n), \hat{P}_{b,n} + c_\delta(n)),$$

where $c_\delta(n) = \sqrt{(1/n)log(1/\delta)}$, and δ is the confidence that $P(b \succ b')$ lies in $\hat{C}_{b,n}$. The function $c_\delta(n)$ decreases as the number of comparisons n increases. By properly setting parameter δ, the optimal reward can be reached within the fixed time horizon.

Active-Elimination (Algorithm 3) is the key part of *Rank-Comparison*. For each group of tests, d arms are randomly chosen from W_ℓ with no repeats when $d < |W_\ell|$. Otherwise, we pick each arm equally and pick the rest arms randomly according to lines 7-9 in Algorithm 3. The randomized selection method provides low-variance total regret. Each group of tests results in a ranking of d arms, which can be regarded as $d(d-1)/2$ comparisons among the d arms. For each arm b, the values of w_b, n_b and $\hat{P}_b$ are updated, as is the corresponding confidence radius c^*. For any pair of arms b and b', one dominates the other if their confidence intervals do not overlap, and the less superior arm is eliminated from W_ℓ. The algorithm runs until the time horizon $T = d \cdot G$ is reached, or until only one active arm remains.

4.4 Theoretical Results

The patients can rank at most, the performances of d stimuli responses. For fixed time horizon T, choose the size of the groups equals to the maximum group size d. It will maximize the number of total comparisons extracted from the ranks, which is $d(d-1)/2$.

Let $\epsilon = \epsilon_{1,2}$ to be the comparison factor between the best and second best arms. Obviously, we have $\epsilon \leq \epsilon_{1,j}$ for all j. The upper bound of the expected total regret for *Rank-Comparison* is given in the theorem below.

Theorem 3 *The expected regret generated by running Algorithm 1 is bounded from above by $O(\frac{K}{\epsilon \cdot d} logT)$.*

As compared to the classical dueling bandit regret bound of $O(\frac{K}{\epsilon} logT)$, *Rank-Comparison* has an extra divisor factor of d. This tighter bound is realized because for each group of d tests, order $O(d^2)$ comparisons are extracted from the ranking test. Recall that $R_T = 0$ if the optimal arm $b_{(t)} = b_1$ is constantly chosen, and $R_T = \Theta(T)$ is linear $w.r.t.$ T if we constantly choose $b_{(t)} \in \{b_1, \cdots, b_K\}$. The factor $O(\frac{K}{\epsilon \cdot d} logT)$ lies in the region between 0 and $\Theta(T)$. As T increases, $O(\frac{K}{\epsilon \cdot d} logT)$ is significantly less than $\Theta(T)$.

By extending Theorem 4 of Yue et al. (2009), we can form a lower bound on regret in expectation, as stated in Theorem 2, for any algorithm which solves the rank comparison problem, which means no algorithm can achieve lower regret than *Rank-Comparison* in expectation.

Theorem 4 *Any algorithm for the rank comparison problem has a regret bounded from below by $\Omega(\frac{K}{\epsilon \cdot d} logT)$.*

Notice that Theorem 4 lower bounds total regret on the same order as the upper bound in Theorem 1. So we have $\Omega(\frac{K}{\epsilon \cdot d} logT) = E[R_T] = O(\frac{K}{\epsilon \cdot d} logT)$, from which we can conclude that total regret is order $\Theta(\frac{K}{\epsilon \cdot d} logT)$ for *Rank-Comparison*.

Theorems 3 and 4, whose detailed proofs can be found in the supplementary of Sui and J. Burdick (2014a), show that our algorithm is optimal in terms of the expected total regret.

Unlike the classical multi-armed bandit problem, which only focuses on expected total regret, many applications must constrain the regret's variation. In our context, if a stimulus optimization algorithm provides good results in the majority of patients, but bad results in a few, the variation is large. Such an algorithm is not practically useful, even if total regret is small. By randomizing the choice of arms within each test group, the randomized comparison strategy of *Rank-Comparison* provides low-variance regret in expectation.

4.5 Experiments

We first evaluate the algorithm by simulation. The reward for each arm b_i is modeled as a Gaussian distribution with mean μ_i and standard deviation σ_i. All arms are independent with each other. Obviously, the distributions generated in this way satisfies the Strong Stochastic Transitivity and Stochastic Triangle Inequality. Then we sample the arms for each group and rank them by using the Rank-Comparison algorithm. We calculated the expected regret $r_t = R_t/t$ (instead of total regret R_t) where t is the number of tests. In the simulation, we consider the total number of arms is 10 and we can get rank list with dimension no larger than 5. The reward of each arm b_i follows a Gaussian distribution with mean $\mu_i \in [0, 1]$ and standard deviation $\sigma_i = 0.2$. Set the confidence parameter $\delta = 10^{-2}$.

Under this setting, the arms are hard to be distinguished from each other due to the large variances. Figure 4.2 shows the mean regret r_t vs. time t for *Rank-Comparison* (blue curve) and Beat-the-mean (Yue and Joachims, 2011) (red curve) with fixed horizon $T = 1000$. The blue curve is the mean regret of *Rank-Comparison*, while the red curve is the mean regret of Beat-the-Mean algorithm. For both algorithms, the mean regret is high during exploration, and then drops quickly after the algorithms converge to the optimum. We can see that *Rank-Comparison* finds the optimum within 150 tests, and thereafter exploits it to reduce the mean regret. However, Beat-the-Mean did not converge to the optimum within the time horizon for the same parameter settings. We hypothesize that *Rank-Comparison* outperforms Beat-

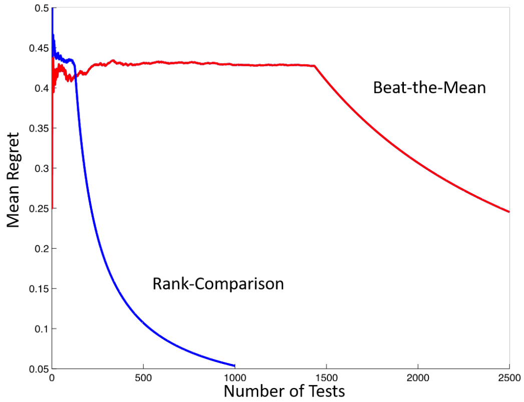


Figure 4.2: Mean Regret against Number of Tests

the-Mean because of the utility of finer feedback information.

4.6 Discussion and Conclusion

This chapter proposed a *Rank-Comparison* algorithm to efficiently solve a specific bandit problem using subgroup rank feedback. This optimal strategy (Theorems 3 and 4) provides clinical recommendation which explores for optimal stimuli while exploiting high performing stimuli for SCI therapy. The main advantages of *Rank-Comparison* are:

- Faster convergence, which is a necessity for applications which are characterized by expensive explorations.
- Low variance of the reward/regret (R_T), which guarantees that the approach performs uniformly on the majority of patients.

Rank-Comparison decomposes test group rankings into equally weighted comparisons. One might reasonably assume that arms far apart in rank may be more distinguishable than adjacent ones, and thus employ different confidence parameters as appropriate. This feature can reduce total regret under the same problem setting. From the clinical point of view, this method avoids the varying effect of human judgement by using robust comparisons instead of volatile quantitative values, which may be non-stationary in our application. However, the time varying characteristics of human motor performance due to fatigue in the short term, and spinal plasticity over the long term, is a real theoretical and clinical issue we must address.

Additionally, the classical bandit problem's assumption of independent arms does not hold for the spinal cord stimulation where anatomical principles and electrical properties suggest a coupling occurs. Using a measure of similarity between stimuli based on the physical properties, we can build a prior distribution on unknown arms to guide our search.

Chapter 5

THEORETICAL CONTRIBUTIONS: SAFE EXPLORATION

Usually, clinical experiments were performed having a human researcher to control the stimulator and the recording system, while the algorithm performed an executive or directing role. This architecture has a number of advantages, among them that the human experimenter provides a fail-safe with respect to data acquisition (e.g., if an element of the data processing fails, the observations of the human experimenter can often be used to reconstruct the missing information) as well as with respect to safety (if an unexpected condition arises, the human experimenter can terminate stimulation, or, if a stimulus known to be painful is requested, the human experimenter can refuse to perform the requested experiment). This latter fail safe must be maintained in an automatic system.

In many applications, the guarantee of safety is crucial to the learning algorithm. This chapter studies how to safely sample the input space towards global optimum with real valued feedback. In this chapter, we consider sequential decision problems under uncertainty, where we seek to optimize an unknown function from noisy samples. This requires balancing exploration (learning about the objective) and exploitation (localizing the maximum), a problem well-studied in the multi-armed bandit literature. In many applications, however, we require that the sampled function values exceed some pre-specified “safety” threshold, a requirement that existing algorithms fail to meet. Examples include medical applications where the patients’ comfort must be guaranteed; recommender systems aiming to avoid user dissatisfaction; and robotic control, where one seeks to avoid controls causing physical harm to the platform. We tackle this novel, yet rich, set of problems under the assumption that the unknown function satisfies regularity conditions expressed via a Gaussian process prior. We develop an efficient algorithm called SafeOpt, and theoretically guarantee its convergence to a natural notion of optimum reachable under safety constraints. We extensively evaluate SafeOpt on synthetic data, as well as two real applications: movie recommendation, and therapeutic spinal cord stimulation.

5.1 Introduction

Many applications in recommender system or experimental design need to make decisions online. Each decision leads to a stochastic reward with initially unknown distribution, while new decisions are made based on the observations of previous rewards. To maximize the total reward, one needs to solve the tradeoff between exploring different strategies and exploiting currently optimal strategies within a given set of strategies. However, when the sample space (set of strategies) is not free to explore, classical bandit algorithms are no longer optimal. In some applications, it is unacceptable to ever incur low rewards; it is required that the reward of any sampled strategy be above some specified "safety" threshold.

We are aiming to sequentially optimize an unknown reward function while preventing to sample at low-rewarding places in the sample space. A threshold is set to prevent low-rewarding sampling. Often, we have some known samples above the threshold before we start the optimization.

Consider, for example, medical applications (e.g., in rehabilitation), where physicians may choose among a large set of therapies. The effects of different therapies are initially unknown and can only be determined through experimentation. Free exploration, however, is not possible, since some therapies might cause severe discomfort or even physical harm to the patient. Oftentimes, the effects of similar therapies are correlated. Therefore, a feasible way to explore might be to start from some therapies similar to those known to be safe, since their efficacy would not be too different from the known ones. This way, more and more choices can be established to be safe, facilitating further exploration. In Section 5.5, we address an instance of such a problem, with the goal of choosing stimulation patterns (therapies) for epidurally implanted electrode arrays to aid rehabilitation of patients that have suffered spinal cord injuries. Similar challenges arise in robotic control, where we aim to learn a controller by experimenting with the robot, yet some parameters might lead to physical harm to the platform. The problem also arises in domains like recommender systems, where we might wish to avoid recommendations that are severely disliked by the user, an application we also consider in Section 5.5.

Related work. The tradeoff between exploration and exploitation is classically studied in context of the (stochastic) multi-armed bandit problem. It models sequential decision tasks in which one chooses among a number of different decisions (arms), each associated with a stochastic reward with initially unknown distribution.

The goal of a bandit algorithm is to maximize the cumulative reward. In a variant called “best-arm identification” (Audibert, Sebastien Bubeck, and Remi Munos, 2010), one seeks to identify the decision with highest reward with minimal trials. Since its introduction by Robbins (1952), bandit problems have been widely studied in many situations (cf., Sébastien Bubeck and Cesa-Bianchi (2012) for an overview). Many efficient algorithms build on the work of Auer (2002), and their key idea is to use *upper confidence bounds* to implicitly negotiate the explore-exploit tradeoff by optimistic sampling. This idea naturally extends to bandit problems with complex (or even infinite) decision sets under certain regularity conditions of the reward function (Dani, Hayes, and Kakade, 2008; Kleinberg, Slivkins, and Upfal, 2008; Sébastien Bubeck, Rémi Munos, et al., 2008). Srinivas et al. (2010a) show how confidence bounds can be used to address bandit problems with a reward function that is modeled using a Gaussian process (GP), a regularity assumption also commonly made in Bayesian optimization (c.f., (Brochu, Cora, and Freitas, 2010)), which is closely related to best-arm identification. These approaches effectively optimize long-term performance by accepting low immediate rewards for sake of exploration. While this compromise is acceptable in certain settings, it makes these techniques unsuitable in safety-critical applications. In particular, the GP-UCB algorithm uses GP-inferred upper confidence bounds for selecting samples and has been shown to achieve sub-linear regret. GP-UCB is an effective method to tackle the global optimization problem. However, if we need to determine the set of points, for which the function takes value above some given threshold level and restricts the new sample points to be chosen only within the set which has function values above the threshold, GP-UCB is no longer feasible. Running it leads us to points with function values below threshold, which is not acceptable. Another problem that has been studied, is that of active sampling for localizing level sets, that is, decisions where the objective crosses a specified threshold (Bryan et al., 2005; Gotovos et al., 2013). However, these approaches generally sample both above and below the threshold, which makes them also unsuitable for safety-critical settings.

The problem of safe exploration has been considered in control and reinforcement learning (Hans, Schäfer, and Udluft, 2008; Gillula and Tomlin, 2011; Garcia and Fernandez, 2012). For example, Moldovan and Abbeel (2012) consider the problem of safe exploration in MDPs. They ensure safety by restricting policies to be ergodic with high probability, i.e., able to “recover” from any state visited. This is a more general problem, which comes at a cost—feasible safe policies do not always exist, algorithms are far more complex, and there are no convergence guarantees. In contrast, we restrict ourselves to the bandit/optimization setting, where decisions do not cause state transitions, which leads to simpler algorithms with stronger guarantees, even in the agnostic (non-Bayesian) setting.

Our contributions. We model a novel class of safe optimization problems as maximizing an unknown expected-reward function over the decision set from noisy samples. By exploiting regularity conditions on the function, which capture the intuition that similar decisions are associated with similar rewards, we aim to balance exploration (learning about the function) and exploitation (identifying near-optimal decisions). The requirement to ensure safety leads to novel considerations, different from those addressed in bandits, where we must not only explore to reduce uncertainty about the function, but also expand the set of decisions established as safe.

Concretely, we propose a novel algorithm, SafeOpt, to balance this tradeoff. SafeOpt models the unknown function as a sample from a Gaussian process (GP) prior, and uses the predictive uncertainty to guide exploration. In particular, it uses confidence bounds to assess safety of as yet unexplored decisions. We theoretically analyze SafeOpt under the assumptions that (1) the objective has bounded norm in the Reproducing Kernel Hilbert Space associated with the GP covariance function, and (2) the objective is Lipschitz-continuous, which is guaranteed by many common kernels. We establish convergence of SafeOpt to a natural notion of "safely reachable" optimum decision. We further extensively evaluate SafeOpt on two real-world applications: movie recommendation, and therapeutic stimulation of patients with spinal cord injuries.

5.2 Problem Statement

We consider a sequential decision problem, where we seek to optimize from noisy samples an unknown reward function $f : D \to \mathbb{R}$ defined on a finite set of decisions D. Concretely, we pick a sequence of decisions (e.g., items to recommend, experimental stimuli) $x_1, x_2, \dots \in D$, and, after each selection x_t, get back a noise-perturbed value of f, that is, we observe $y_t = f(x_t) + n_t$ (e.g., user rating, stimulus response). Our goal is to identify a decision x^* of maximum reward f, akin to the problem of best-arm identification in multi-armed bandits (Audibert, Sebastien Bubeck, and Remi Munos, 2010). As crucial difference, however, we wish to ensure that, for all rounds t, it holds that $f(x_t) \geq h$ with high probability, where h is a problem-specific parameter. We call decisions for which $f(x_t) \geq h$ *safe*. In our recommender systems example, this means that we seek to identify items with utility to the user, while guaranteeing that we never propose items the user strongly dislikes. In our rehabilitation setting, we seek to guarantee that no painful stimuli are applied. Importantly, since f is unknown, the set of safe decisions is initially unknown as well.

Regularity assumptions. Without assumptions, this is clearly a hopeless task. In particular, without any knowledge of f, we do not even know where to start our exploration. Therefore, we assume that, before starting the optimization, we know a "seed" set of at least one safe decision, which we denote by $S_0 \subset D$. This establishes starting points for our exploration. Without further assumptions on f, we would never be able to identify new safe decisions to consider for exploration. In what follows, we assume that D is endowed with a positive definite kernel function, and that function f has bounded norm in the associated Reproducing Kernel Hilbert Space (RKHS, cf., Schölkopf and Smola (2002)). Note that for finite decision sets and any universal kernel, this assumption is automatically satisfied. This assumption allows us to model our reward function f as a sample from a Gaussian process (GP) (Rasmussen and Williams, 2006). A $GP(\mu(x), k(x, x'))$ is a probability distribution across a class of "smooth" functions, which is parameterized by a kernel function $k(x, x')$ that characterizes the smoothness of f. We assume w.l.o.g. that $\mu(x) = 0$, and that our observations are perturbed by i.i.d. Gaussian noise, i.e., for samples at points $A_T = [x_1 \dots x_T]^T \subseteq D$, we have $y_t = f(x_t) + n_t$ where n_t $N(0, \sigma^2)$. (We will relax this assumption later.) The posterior over f is then also Gaussian with mean $\mu_T(x)$, covariance $k_T(x, x')$, and variance $\sigma_T^2(x, x')$ that satisfy,

$$\begin{aligned}
\mu_T(x) &= k_T(x)^T (K_T + \sigma^2 I)^{-1} y_T \\
k_T(x, x') &= k(x, x') - k_T(x)^T (K_T + \sigma^2 I)^{-1} k_T(x') \\
\sigma_T^2(x) &= k_T(x, x),
\end{aligned}$$

where $k_T(x) = [k(x_1, x) \dots k(x_T, x)]^T$ and K_T is the positive definite kernel matrix $[k(x, x')]_{x, x' \in A_T}$.

Why should this assumption help? The predictive confidence of the GP posterior will allow us to reason about which points in D are safe with high probability. For sake of our exposition and analysis, we will further assume that f is L-Lipschitz continuous w.r.t. some metric d on D. This is automatically satisfied, for example, when considering commonly used isotropic kernels, such as the Gaussian kernel, on D.

Optimization goal. Under the Lipschitz-continuity assumption, what is the best solution that *any* algorithm might be able to find? Suppose our observations were noise free. In this case, after exploring the decisions in our seed set S_0, we can establish any decision x as safe, if there exists a decision $x' \in S_0$, such that $f(\times x') - L \cdot d(x, x') \geq h$. Exploring these newly identified safe decisions will establish further decisions as safe, and so on. Unfortunately our knowledge of f comes from noisy observations, so even after experimenting with the same decision $\times$ x repeatedly, we are not able to infer $f(x)$ exactly, but only up to some statistical confidence $f(x) \pm \epsilon$. Based on this insight, we define the *one-step reachability* operator

$$R_\epsilon(S) := S \cup \left\{x \in D \;\middle|\; \exists x' \in S, f(x') - \epsilon - Ld(x', x) \geq h\right\},$$

which represents the subset of D that can be established as safe upon learning f up to absolute error at most ϵ within S. Clearly, it holds that $S \subseteq R_\epsilon(S) \subseteq D$. Similarly, we can define the n-step reachability operator by

$$R^n_\epsilon(S) := \underbrace{R_\epsilon(R_\epsilon \dots (R_\epsilon(S))}_{n \text{ times}} \dots),$$

and its closure by $\bar{R}_\epsilon(S) := \lim_{n \to \infty} R^n_\epsilon(S)$. It is easy to see that *no* algorithm that is able to learn f only up to ϵ will ever be able to establish any $x \in D \setminus \bar{R}_\epsilon(S_0)$ as safe. Therefore, we cannot hope that any safe algorithm will be able to identify the global optimum $f^* = \max_{x \in D} f(x)$. We consider, instead, our benchmark to be the ϵ-reachable maximum

$$f^*_\epsilon = \max_{x \in \bar{R}_\epsilon(S_0)} f(x). \tag{5.1}$$

We know $\bar{R}_\epsilon(S_0)$ is the largest subset we could explore in D. Our real problem is to optimize the function within $\bar{R}_\epsilon(S_0)$ instead of D to guarantee ϵ-safe. However, we do not know $\bar{R}_\epsilon(S_0)$ during the sampling process. We can only reach the sample space S_t which is a subset of $R^t_\epsilon(S_0)$ at iteration t. Starting from the seed set S_0, we need to optimize the reward function within the safe region and expand the safe region towards $\bar{R}_\epsilon(S_0)$ in the meantime.

From Lipschitz continuity, we can infer a safe set $S_1 \subset D$ which contains all the points in D that have a guarantee that their sampled value will be above the threshold h. We choose a point $x_1 \in S_1$ and get the function value perturbed by noise there: $y_1 = f(x_1) + n_1$ where n_1 represents the noise at this sample. We have known $y_1 > h$ since x_1 is sampled from the current safe set. If the sampled value has high function value or locates near the boundary of the current safe set, it has the potential to enlarge the safe set. We define

$$g_t(x) := \left|\left\{x' \in D \setminus S_t \,\middle|\, u_t(x) - Ld(x, x') \geq h\right\}\right|$$

to be the cardinality of the enlargement of the current safe set after we sample a new point x. So we after each iteration, we have a new safe region S_t from which the next sample point will be chosen.

Since we need to guarantee not sample at low-valued points, our sampling decisions always be made within the safe set S_t which is a subset of the whole space D. Obviously, S_t is a non-decreasing sequence. So what is the largest set S_t we could reach when $t \longrightarrow \infty$ given the initial seed set S_0? We first enroll the idea of ϵ-safe point x by $f(x') - \epsilon - Ld(x', x) \geq h$, where ϵ is the safety margin to represent how far the value at point x from above the threshold h. Larger margin ϵ means better safety guarantee at point x.

Failure of naive approaches. There are a number of approaches for trading exploration and exploitation under the smoothness assumptions expressed via a GP. One such approach is the *Gaussian Process Upper Confidence Bound* algorithm (GP-UCB), which greedily chooses

$$x_t = \operatorname*{argmax}_{x \in D} \mu_{t-1}(x) + \beta_t^{1/2} \sigma_{t-1}(x) \tag{5.2}$$

for a suitable schedule of β_t. While this algorithm is guaranteed to achieve sublinear cumulative regret (Srinivas et al., 2010a), it places no restrictions on the sampling location, and hence neither theoretically guarantees safety, nor exhibits it in our experiments. This is symptomatic of typical multi-armed bandit approaches when applied to our problem. In the following, we will present an efficient algorithm, SafeOpt, which, under the aforementioned assumptions, is guaranteed, for any $\epsilon > 0$ and $\delta > 0$, to identify a solution $\hat{x}$, such that $f(\hat{x}) \geq f_\epsilon^* - \epsilon$, with probability at least $1 - \delta$. In Section 5.4, we will further provide a sample complexity bound on the number of iterations required to achieve this condition.

5.3 SafeOpt Algorithm

We now introduce our proposed algorithm, SafeOpt, for the safe exploration for optimization problem.

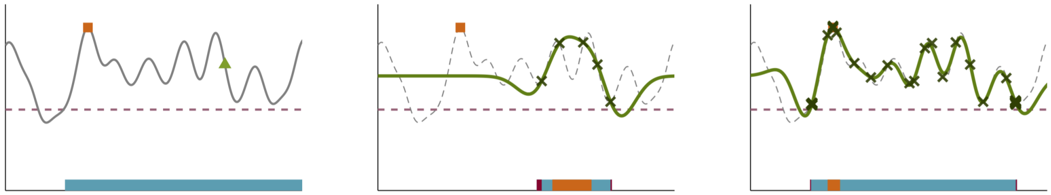


Figure 5.1: Illustration of SafeOpt. (a) The solid curve is the (unknown) function to optimize. The straight dashed line represents the threshold. The triangle is the safe seed S_0. The cyan bar shows the maximal safe region $\bar{R}_0(S_0)$ reachable w.r.t. the seed set S_0 and the square is the optimum within this safe region. (b, c) The solid line is the estimated mean function after a number of observations, indicated by crosses. The expansion set G_t is shown as the dark purple bars on the two ends of S_t, and the set of candidate maximizers M_t is shown as the orange bar within S_t.

Overview. We start with a high-level description of SafeOpt. The algorithm uses Gaussian processes to make predictions about f based on noisy evaluations, and uses their predictive uncertainty to guide exploration. To guarantee safety, it maintains an increasing sequence of subsets $S_t \subseteq D$ established as safe using the GP posterior. It never chooses a sample outside of set S_t, while it balances two objectives within that set: the desire to explore by expanding the safe region, and the need to localize high-reward regions within S_t. For the former, it maintains a set $G_t \subseteq S_t$ of candidate decisions that – upon potentially repeated selection – have a chance to expand the reachable region. For the latter, it maintains a set $M_t \subseteq S_t$ of decisions that are potential maximizers of f. To make progress, in each round it greedily picks the most uncertain decision x, that is, the one with largest predictive variance among $G_t \cup M_t$. We present pseudo-code of SafeOpt in Algorithm 11, and next explain its workings in more detail.

Confidence-based classification. The classification of the domain into sets M_t, G_t, and S_t is done according to the GP posterior. In particular, in iteration t, it uses the predictive confidence intervals

$$Q_t(x) \coloneqq \left[\mu_{t-1}(x) \pm \beta_t^{1/2}\sigma_{t-1}(x)\right], \tag{5.3}$$

where β_t is defined in Theorem 5. Based on the assumptions about f, the sampled reward value at x lies in $Q_t(x)$ with high probability for all t. For technical reasons, instead of using Q_t directly, we use their intersection $C_t(x) \coloneqq C_{t-1}(x) \cap Q_t(x)$, which ensures that confidence intervals are monotonically contained in each other. Based on this notion, we define $u_t(x) \coloneqq \max_{x \in D} C_t(x)$ as a high-probability upper confidence bound on $f(x)$, monotonically decreasing in t, and similarly, $\ell_t(x) := \min_{x \in D} C_t(x)$ as a lower confidence bound, monotonically increasing in t. We also define the width $w_t(x) := u_t(x) - \ell_t(x)$ of the confidence interval, which is monotonically decreasing in t and captures the uncertainty of the GP model about decision x.

Having introduced the above notation, we define the essential sets of our algorithm, S_t, M_t, and G_t. The decisions that are certified to be safe are given by the set

$$S_t = \bigcup_{x \in S_{t-1}} \left\{x' \in D \;\middle|\; \ell_t(x) - Ld(x, x') \geq h\right\}.$$

The potential maximizers are those decisions, for which the upper confidence bound is higher than the largest lower confidence bound, i.e.,

$$M_t = \{x \in S_t : u_t(x) \geq \max_{x' \in S_t} \ell_t(x')\}.$$

In order to identify the set G_t, we first define the function

$$g_t(x) := \left|\left\{x' \in D \setminus S_t \;\middle|\; u_t(x) - Ld(x, x') \geq h\right\}\right|,$$

which (optimistically) quantifies the potential enlargement of the current safe set after we sample a new decision x. Then, G_t is simply given by

$$G_t = \{x \in S_t : g_t(x) > 0\}.$$

Sampling criterion. Given the classification of points presented above, the selection rule is very simple: SafeOpt just greedily selects the most uncertain decision that could either be a maximizer (in M_t), or enlarge the reachable region (in G_t). Formally, it selects decision x_t by

$$x_t \in \operatorname*{argmax}_{x \in M_t \cup G_t} w_t(x).$$

Reducing the uncertainty within G_t will eventually lead to expansion, i.e., the discovery of new safe decisions. In turn, sampling within M_t will reduce the uncertainty about the location of f's maximum within S_t. The greedy selection balances these two goals. An illustration of the sampling process is shown in Figure 5.1. We start given a single seed decision and the initial singleton safe set S_0 it generates. After several iterations, the safe set S_t grows, while SafeOpt picks new points from $G_t \cup M_t$. After a large number of samples, $G_t \cup M_t$ shrinks towards the empty set and S_t converges toward the total safe region also finding a near-optimal decision within it.

Discussion. The sets S_t, G_t, and M_t exhibit some interesting dynamics. As mentioned above, the reachable region S_t is monotonically increasing, i.e., $S_0 \subseteq S_1 \subseteq S_2 \ldots$. The algorithm proceeds in stages, within each of which the set S_t does not change (i.e., $S_t = S_{t+1}$). This is because not enough evidence has been accrued yet to establish new decisions as safe. Within each such stage, the sets G_t and M_t keep shrinking, due to the monotonicity of the confidence bounds used. However, as soon as new decisions are identified as safe, G_t and M_t may increase again. Furthermore, note that, even though we defined the optimization goal (5.1) with respect to some accuracy parameter ϵ, this parameter is actually not used by the algorithm, although it can be employed as a stopping condition. Namely, if the algorithm stops under the following condition,

$$\max_{x \in M_t \cup G_t} w_t(x) \leq \epsilon,$$

then for the point $\hat{x} = \text{argmax}_{x \in S_t} \ell_t(x)$ it holds that $f(\hat{x}) \geq f^*_\epsilon - \varepsilon$.

The safe/accuracy parameter ϵ does not effect the iteration process. It provides a guarantee of accuracy after we finish some iterations. So for the infinite horizon case, there is no need to set that ϵ beforehand.

We also considered an safely search version of GP-UCB which we call it Local-UCB in Algorithm 12. Comparing to the GP-UCB which achieves the global optimization problem, Local-UCB use the same combined strategy to choose x_t:

$$x_t = \underset{x \in S_t}{\text{argmax}} (\mu_{t-1}(x) + \beta_t^{1/2} \sigma_{t-1}(x)).$$

The main difference is to choose a sample point from safe region S_t instead of total space D. Also, S_t will be updated for each iteration.

5.4 Theoretical Results

We now establish the effectiveness of SafeOpt by theoretically bounding its sample complexity. The two critical behaviors of SafeOpt are the expansion of the safe region in search for the total safe region, and the optimization within the safe region.

Accuracy of confidence sets. The correctness of SafeOpt crucially relies on the fact that the classification into sets S_t, M_t, and G_t is accurate. While this requires that the confidence bounds C_t are conservative, using bounds that are too conservative will slow down the algorithm considerably. Tightness of the confidence bounds is controlled by parameter β_t in equation (5.3), the choice of which is crucial. This problem of properly tuning confidence bounds in exploration–exploitation tradeoffs

Algorithm 11 SafeOpt

```
Input: sample set $D$,
       GP prior $(\mu_0, k, \sigma_0)$,
       Lipschitz constant $L$,
       seed set $S_0$,
       safe threshold $h$        accuracy $\epsilon$
$C_0(x) \leftarrow [h, \infty)$, for all $x \in S_0$
$C_0(x) \leftarrow \mathbb{R}$, for all $x \in D \setminus S_0$
$Q_0(x) \leftarrow \mathbb{R}$, for all $x \in D$
for $t = 1, \dots$ do
    $C_t(x) \leftarrow C_{t-1}(x) \cap Q_{t-1}(x)$
    $S_t \leftarrow \bigcup_{x \in S_{t-1}} \left\{ x' \in D \,\middle|\, \ell_t(x) - Ld(x, x') \geq h \right\}$
    $G_t \leftarrow \left\{ x \in S_t \,\middle|\, g_t(x) > 0 \right\}$
    $M_t \leftarrow \left\{ x \in S_t \,\middle|\, u_t(x) \geq \max_{x' \in S_t} \ell_t(x') \right\}$
    $x_t \leftarrow \operatorname{argmax}_{x \in G_t \cup M_t} (w_t(x))$
    $y_t \leftarrow f(x_t) + n_t$
    Compute $Q_t(x)$, for all $x \in S_t$
end for
```

Algorithm 12 Local-UCB

```
Input: sample set $D$,
       GP prior $(\mu_0, k, \sigma_0)$,
       seed set $S_0$,
       safe threshold $h$,
       accuracy $\epsilon$
for $t = 1, \dots$ do
    $S_t \leftarrow \bigcup_{x \in S_{t-1}} \left\{ x' \in D \,\middle|\, \ell_t(x) - Ld(x, x') \geq h \right\}$
    $x_t \leftarrow \operatorname{argmax}_{x \in S_t} (\mu_{t-1}(x) + \beta_t^{1/2} \sigma_{t-1}(x))$
    $y_t \leftarrow f(x_t) + n_t$
end for
```

involving Gaussian processes has been studied by Srinivas et al. (2010a). While they consider the bandit problem, i.e., maximizing average reward, without safety guarantees, we show below that their choice of confidence bounds can be generalized to our setting. In particular, for our theoretical results to hold it suffices to choose

$$\beta_t = 2B + 300\gamma_t \log^3(t/\delta), \tag{5.4}$$

where B is a bound on the RKHS norm of f, δ is the allowed failure probability, and γ_t quantifies the effective degrees of freedom associated with the kernel function. Concretely,

$$\gamma_t = \max_{|A| \leq t} I(f; y_A)$$

is the maximal mutual information that can be obtained about the GP prior from t samples. For finite $|D|$, this quantity is always bounded by $\gamma_t \leq |D| \log(1 + \sigma^{-2} t |D| \max_{x \in D} k(x,x))$, i.e., $O(|D| \log t |D|)$, but for commonly used kernels (such as the Gaussian kernel), γ_t has sublinear dependence on $|D|$ (Srinivas et al., 2010a).

Lemma 7 *Suppose $\|f\|_k^2 \leq B$, and suppose the noise n_t is zero-mean conditioned on the history, and uniformly bounded by σ for all t. Select β_t as in* (5.4). *Then, with probability at least* $1 - \delta$*, for all iterations t during the execution of SafeOpt and for all $x \in D$ it holds that $f(x) \in C_t(x)$.*

Proof. This lemma immediately follows from Theorem 6 of Srinivas et al. (2010a), and our construction of the confidence sets C_t.

Convergence of SafeOpt. Given this result, we now present our main theorem, which establishes that SafeOpt indeed manages to identify an ϵ-optimal decision, while staying safe throughout.

Theorem 5 *Assume that f satisfies $\|f\|_k^2 \leq B$ and f further is L-Lipschitz continuous. Also, suppose $S_0 \neq \emptyset$, and $f(x) \geq h$, for all $x \in S_0$. Fix any $\epsilon > 0$ and $\delta \in (0,1)$. Suppose we run SafeOpt with seed set S_0 under the same assumptions on the noise n_t and same choice of β_t as in Lemma 7. Define $\hat{x}_t = \mathrm{argmax}_{x \in S_t} \ell_t(x)$ and let t^* be the smallest positive integer satisfying*

$$\frac{t^*}{\beta_{t^*} \gamma_{t^*}} \geq \frac{C_1 \left(|\bar{R}_0(S_0)| + 1 \right)}{\epsilon^2},$$

where $C_1 = 8/\log(1 + \sigma^{-2})$. Then, the following jointly hold with probability at least $1 - \delta$*:*

- $\forall t \geq 1,\ f(x_t) \geq h,$
- $\forall t \geq t^*,\ f(\hat{x}_t) \geq f^*_\epsilon - \epsilon.$

The detailed proof of Theorem 5 is presented in the Appendix of this chapter. It shows that with high probability, SafeOpt guarantees safety, and identifies at least one ϵ-optimal decision among the ϵ-reachable set after at most t^* iterations. The size of t^* depends on the largest size of safe region $\bar{R}_0(S_0)$, the accuracy parameter ϵ, the confidence parameter δ, the complexity of the function B and the parameterization

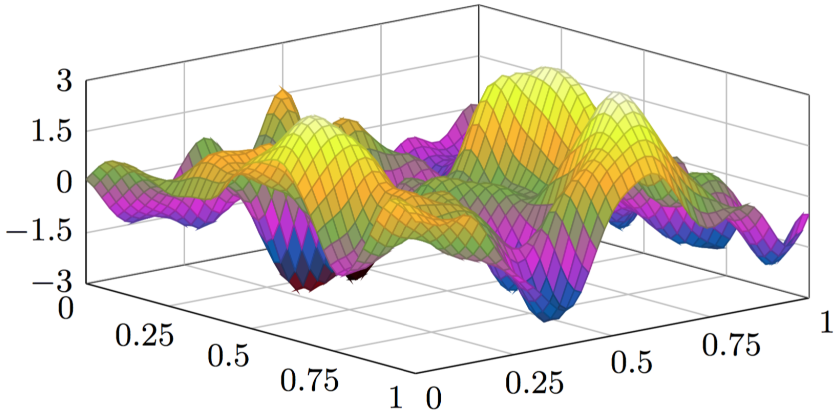


Figure 5.2: A 2-D function sampled from a GP

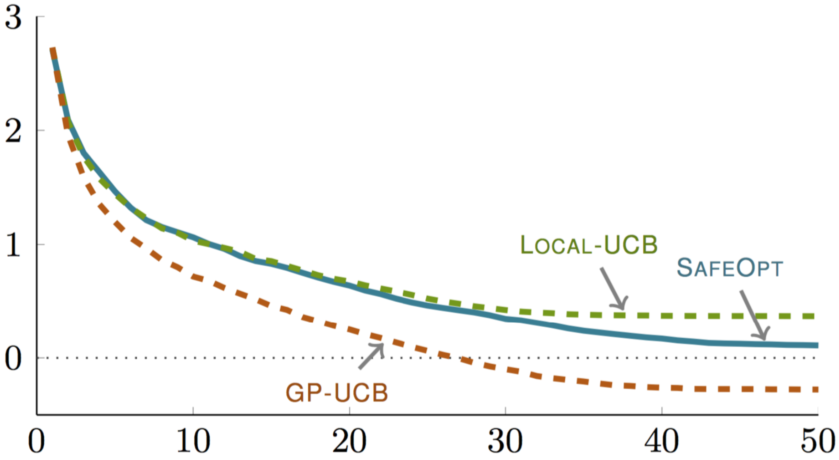


Figure 5.3: The regret of the three algorithms SafeOpt (solid line), Local-UCB (green dashed line) and GP-UCB (red dashed line) v.s. number of iterations.

of the GP via γ_t. The proof is based on the following idea. Within a stage, wherein S_t does not expand, the uncertainty $w_t(x_t)$ monotonically decreases due to construction of M_t and G_t. We prove that the condition $\max_{x \in G_t} w(x) < \epsilon$ implies either of two possibilities: S_t will expand after the next evaluation, i.e., the reachable region will increase, and, therefore, the next stage shall commence; or, we have already established all decisions within $\bar{R}_\epsilon(S_0)$ as safe, i.e., $S_t \subseteq \bar{R}_\epsilon(S_0)$. Similarly, we prove that the condition $\max_{x \in M_t} w(x) < \epsilon$ implies that we have identified an ϵ-optimal decision within the current region S_t. Finally, to establish the sample complexity we use a bound on how quickly $w_t(x_t)$ decreases.

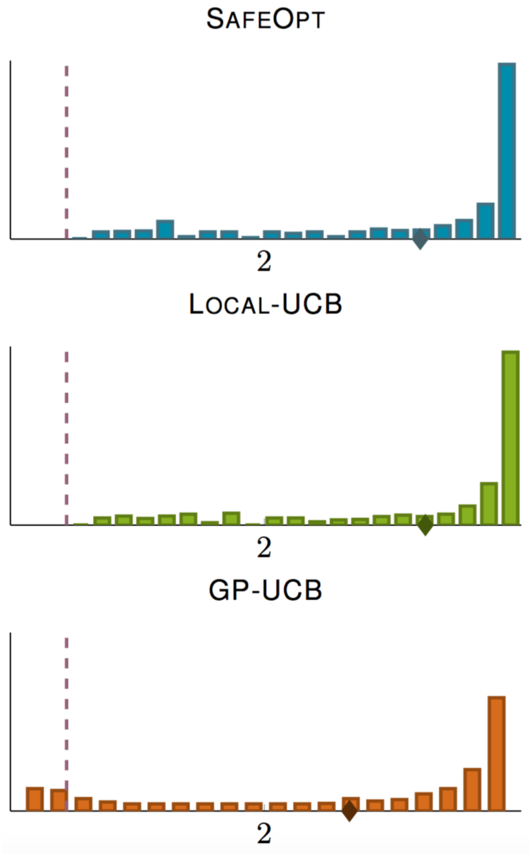


Figure 5.4: Histogram of sampled function values after 50 iterations. The dashed lines represent the threshold and diamonds indicate the mean of the sampled function values.

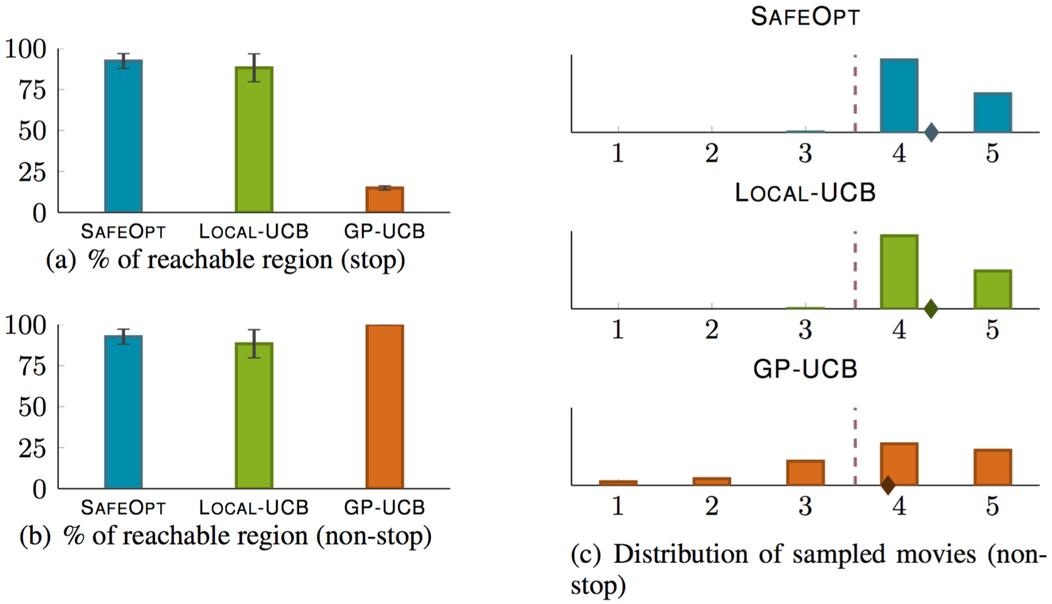
5.5 Experiments

We evaluate our algorithms on synthetic data as well as two real applications. In our experiments, we seek to address the following questions: Does SafeOpt reliably respect the safety requirement? How effective is it in localizing good solutions quickly? How does it compare against standard (non-safe) bandit algorithms? In particular, we compare SafeOpt against GP-UCB (Srinivas et al., 2010a), a multi-armed bandit algorithm designed for Gaussian processes, which however does not respect the safety constraint; and Local-UCB, a heuristic variant of GP-UCB, which selects the sample maximizing the upper confidence bound (similar to GP-UCB), however only among sampling locations that are safe with high probability according to the GP posterior. In particular, it greedily selects

$$x_t = \operatorname*{argmax}_{x \in S_t} \mu_{t-1}(x) + \beta_t^{1/2} \sigma_{t-1}(x).$$

Synthetic data. We first evaluate the algorithm with synthetic data. The purpose of this experiment is to validate our theory, and demonstrate the convergence of SafeOpt in situations that perfectly match our prior assumptions. Concretely, we sampled a set of 100 random functions from a zero-mean Gaussian Process with squared exponential kernel over the sample space $D = [0, 1] \times [0, 1]$, uniformly discretized into 50×50 points. One of the random functions is shown in Figure 5.2. The threshold is set to be the $-\sigma$ plane in the figure. For each random function, we randomly pick 100 safe points, i.e., with values above the threshold. We estimate the Lipschitz constant from the gradient of several random functions sampled from the GP. Regarding each safe point as a separate seed set, we run both Local-UCB and SafeOpt for $T = 50$ iterations. We report a notion of *regret*, defined by $r_t = f_0^* - \max_{1 \leq i \leq t} f(x_i)$. The regret values achieved by each algorithm are averaged over the 100 seeds for each of the 100 random functions. As we can see from Figure 5.3, SafeOpt achieves smaller regret than Local-UCB on average. This is because for some cases Local-UCB fails to expand some low-rewarding boundary points which are slightly above the safe threshold. So it gets stuck at a local optimum. However, SafeOpt balances the localization of the optimal value within the current safe set with the expansion of the reachable region. Since GP-UCB is searching for the global optimum instead of the optimal value within the total safe region, it in fact achieves negative regrets under our definition of regret. Figure 5.4 presents the histogram of the sampled function values obtained by the three algorithms after $T = 50$ iterations. As can be seen, SafeOpt and Local-UCB have very little probability to sample at points below threshold, while a large proportion of points sampled by GP-UCB are unsafe.

Safe movie recommendations. Next we consider an application in recommender systems: how should we recommend movies, while aiming to ensure that the user does not dislike any movies we propose? Concretely, we test the algorithms on the MovieLens-100k dataset, which contains the (sparse) rating of 1682 movies from 943 customers. The main difference between our objective and commonly used objectives such as cumulative reward is that we are not only looking for high scoring movies, but also avoid low scoring ones. To put the problem into our framework, we proceed as follows. We first partition the data by selecting a subset of users for training. On the training data, we apply matrix factorization with $k = 20$ latent factors. This provides a feature vector $\mathbf{v}_i \in \mathbb{R}^k$ for each movie i, and a feature vector $\mathbf{u}_j \in \mathbb{R}^k$ for each user in the training set. We then fit a Gaussian distribution

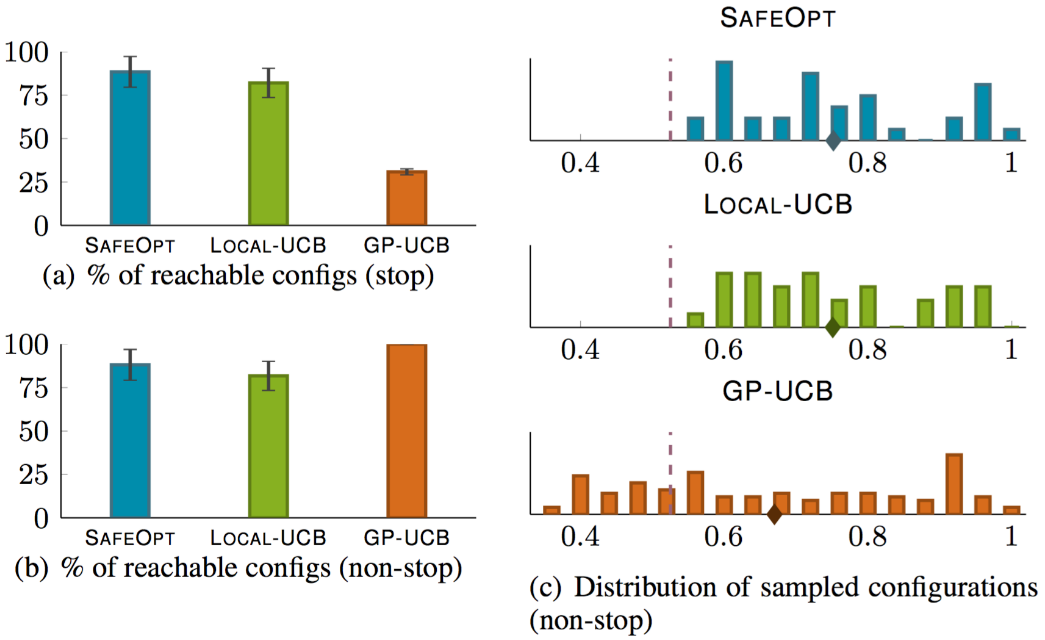


(a) % of reachable region (stop)

(b) % of reachable region (non-stop)

(c) Distribution of sampled movies (non-stop)

Figure 5.5: Safe movie recommendations



(a) % of reachable configs (stop)

(b) % of reachable configs (non-stop)

(c) Distribution of sampled configurations (non-stop)

Figure 5.6: Safe spinal stimulation

$P(\mathbf{u}) = \mathcal{N}(\mathbf{u}; \mu, \Sigma)$ to the training user features. For a new user in the test set, we now consider $P(\mathbf{u})$ as prior, and use the Gaussian likelihood for their ratings for movie $\mathbf{v}_i$ as $P(y_i \mid \mathbf{u}, \mathbf{v}_i) = \mathcal{N}(\mathbf{v}_i^T \mathbf{u}, \sigma^2)$, where σ^2 is the residual variance on the training data. Thus, the ratings $(y_i)_i$ form a Gaussian process (with linear kernel) with Gaussian likelihood.

The safety threshold is set to be the mean of all ratings. We run SafeOpt, Local-UCB together with GP-UCB which is not restricted within the safe region. Given that the dataset only contains partial ratings, we restrict the algorithms only to the movies the user actually rated. After each selection, the actual rating from the data set is provided as feedback to the algorithms. The three algorithms run for $T = 300$ iterations.

The regret reaches zero quickly (within several iterations) since the ratings are discrete from 1 to 5. Figure 5.5(a) shows the percentage of movies explored with respect to the totally reachable set under the constraint that the algorithms will stop after hitting the threshold (i.e., after the first unsafe selection). GP-UCB becomes unsafe and stops much faster than the other two. As a particular example, we see the recommendation for user #5 starting from the known fact that the user rated 5 for the movie “Return of the Jedi”. Within the first 4 iterations, SafeOpt recommends {→ *The Empire Strikes Back* → *Stargate* → *Star Wars* → *Heavy Metal* }, Local-UCB recommends {→ *The Empire Strikes Back* → *Star Wars* → *Star Trek* → *Raiders of the Lost Ark*}; all these movies score above the threshold. GP-UCB recommends {→ *Star Wars* → *Men in Black* → *A Close Shave* → *So I Married an Axe Murderer*}. The last movie recommended by GP-UCB returns a score below the threshold. The movies recommended by SafeOpt share some similarity with those of Local-UCB due to the locality encouraged by safe exploration. GP-UCB on the other hand recommends more diversely since there is no safety restrictions for exploration. Although the average ratings are close, SafeOpt reaches a larger set of movies than Local-UCB because it expands the current safe region more aggressively. Figure 5.5(b) shows the same plots with no stopping criteria applied. GP-UCB exceeds the top ceiling since it can explore beyond the total safe region $\bar{R}_0(S_0)$. It reaches much more movies within the 300 iterations but its mean of ratings is lower and the variance is larger than for the other two algorithms (as in Figure 5.5(c)). This suggests that GP-UCB samples many low-rating movies for exploration. Figure 5.5(c) shows the distribution of ratings of each algorithm. SafeOpt and Local-UCB perform well in preventing the sampling of low-rated movies for each customer.

Safe exploration for spinal cord therapy. Our second application is in a very different domain: spinal cord therapy (Harkema et al., 2011). We compare the algorithms on a dataset of muscle activity triggered by therapeutic spinal electrostimulation in spinal cord injured rats. The clinical goal is to choose stimulating configurations that maximize the resulting activity in lower limb muscles, as measured by electromyography (EMG), in order to improve spinal reflex and locomotor function. Bad configurations have negative effects on the rehabilitation and are often painful. So the configurations we choose must stay above some threshold. We maximize the peak-to-peak amplitude of the recorded EMG waveforms (muscle activities) from the right medial Gastrocnemius muscle. This objective function measures to what degree the selected stimulus activates the interneurons which control reflex activity. Electrode configurations were represented in $\mathbb{R}^4$ by the cathode and anode locations on the array. A squared exponential ARD kernel was fitted for this space using experimental data from 351 stimulations (126 distinct configurations).

Figure 5.6(a) shows the percentages of reachable configurations by the algorithm under the constraint that the algorithms stop after hitting the threshold. Obviously, GP-UCB stops much faster than the other two. SafeOpt reaches larger sets of configurations than Local-UCB. No algorithm dominates on the average activity scores under this stopping criterion. Figure 5.6(b) shows the percentage of reached decisions without stopping upon becoming unsafe. GP-UCB reaches much more configurations, but its mean activity score is lower than the other 2 algorithms (as in Figure 5.6(c)). This indicates that GP-UCB samples many bad configurations. Figure 5.6(c) shows the distribution of muscle activity of each algorithm. SafeOpt and Local-UCB perform well in preventing the sampling of bad configurations.

5.6 Conclusions

We investigated the novel problem of trading exploration and exploitation for function optimization under safety constraints. In particular, we proposed SafeOpt, an efficient algorithm that balances the tradeoff between expanding, exploring, and optimizing over the reachable safe region. We prove strong theoretical performance guarantees for SafeOpt, bounding its sample complexity to achieve an ε-optimal solution while guaranteeing safety with high probability. Our extensive experiments demonstrate that SafeOpt indeed exhibits its analytical safety and convergence properties. We believe that our result provides an important step towards employing machine learning algorithms “live” in safety-critical applications.

Chapter 6

HUMAN EXPERIMENTS: ONLINE LEARNING FOR HUMAN STANDING CONTROL

This chapter provides the online learning algorithm for human standing control and the experimental results with patients. Two patients with severe spinal cord injury were recruited as participants of this study. They satisfy the following conditions:

(1) stable medical condition without cardiopulmonary disease or dysautonomia that would contraindicate standing or stepping with body weight support training;
(2) no painful musculoskeletal dysfunction, unhealed fracture, contracture, pressure sore, or urinary tract infection that might interfere with stand or step training;
(3) no clinically significant depression or ongoing drug abuse;
(4) no current anti-spasticity medication regimen; (5) non-progressive spinal cord injury above T10; (6) AIS A or B;
(7) no motor response present in leg muscles during trans-magnetic stimulation;
(8) not present or bilateral delay of sensory evoked potentials;
(9) no volitional control during voluntary movement attempts in leg muscles as measured by EMG activity;
(10) segmental reflexes remain functional below the lesion;
(11) brain influence on spinal reflexes is not observed as measured by EMG activity;
(12) must not have received Botox injections in the previous 6 months;
(13) be unable to stand or step independently;
(14) at least 1-year post-injury;
(15) must be at least 18 years of age.

Finding the optimal stimulating pattern during the rehabilitation process is modeled as a correlational dueling bandits problem, which is a variant of the dueling bandits problem with the dependence of arms taken into consideration. Many clinical problems with large volume of parameter selection and sequential decision making could be facilitated by this algorithm which makes decisions to simultaneously deliver effective therapy and explore the decision space. We propose an efficient algorithm Correlational Dueling for this problem. After evaluating the convergence of the algorithm in simulation experiments, we apply it to 2 paraplegic subjects implanted with epidural arrays. Experimental results show the effectiveness and efficiency of the algorithm.

6.1 Introduction

In many online learning settings, particularly those that involve human feedback, reliable feedback is often limited to pairwise preferences instead of real valued feedback. Examples include implicit or subjective feedback for information retrieval and recommender systems, such as clicks on search results, and subjective feedback on the quality of recommended care (Chapelle, Joachims, et al., 2012; Sui and J. Burdick, 2014b). This setup motivates the dueling bandits problem (Yue et al., 2012), which formalizes the problem of online regret minimization via preference feedback. (e.g., choosing a pair of arms to be compared at each time step). Many algorithms (Yue and Joachims, 2009; Yue and Joachims, 2011; Zoghi, Whiteson, Remi Munos, et al., 2014; Ailon, Z. Karnin, and Joachims, 2014) have been developed for efficiently computing this problem with independent arms. However, these algorithms are not efficient with many dependent arms. Specifically, when the time horizon T is smaller than the number of arms K, it is hopeless to achieve low regret without utilizing some notion of correlation among arms.

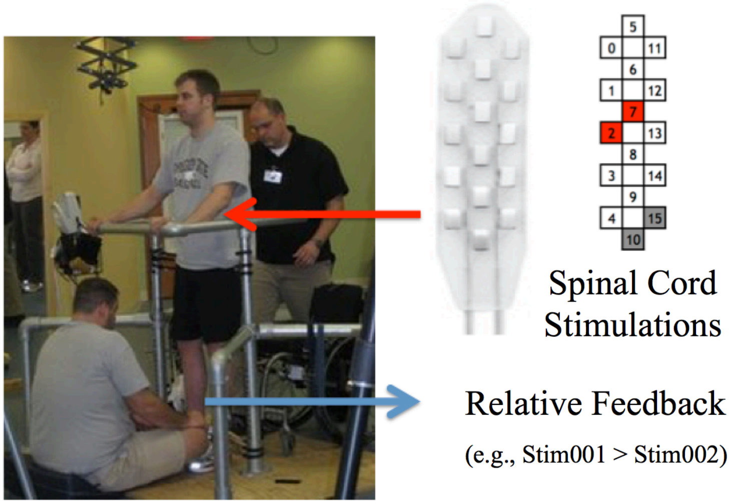


Figure 6.1: The Standing Experiment under spinal stimulation.

Figure 6.1 shows the clinical treatment procedure for *stand-training* under epidural stimulation. During a treatment/optimization session, new stimuli parameters are recommended to be applied and then tested. In the test, the patient attempts to stand using the given stimuli, and the observing clinicians then quantify and compare the patient's resulting performances. Having these noisy comparisons as feedback, we want to continue exploring for the optimal stimulus while also exploiting currently good ones. We must spend significant time dwelling on good performing stimuli in order to provide the patient with a good therapeutic experience. Since clinical training has a fixed time horizon, we must also maximize total performance during the limited period within which we can search for the optimal solution.

We consider the problem of finding optimal stimuli based on the general setting of the multi-armed bandit problem. The classical bandit problem trades off between exploration and exploitation among a number of different arms, each having a quantifiable, but stochastic, reward with initially unknown distribution. However, for the clinical problem, the patient's motor response to stimulation is hard to quantify. Neither video motion capture nor electromyographic (EMG) recordings of muscle activity can yet provide a consistent and satisfactory measure of motor skill under stimulation. A good standing performance might map to numerous combinations of muscle activities, and it is not a stationary process. While the patient's performance under a specific stimulus is hard to quantify, it can be compared to others. In the clinical setting, we can obtain the comparisons of stimuli which are performed within the short time period of one training session.

The *dueling bandit problem* formalizes online learning problems with preference feedback instead of absolute rewards, and hence it can be used for problems with unquantifiable reward. The total number of different stimulating configurations is $\sim 4.3 \times 10^7$, due to the complexity of electrodes. It is not feasible to search through the whole space since the configurations are correlated. The algorithm we propose in this paper is a variant of the dueling bandit algorithm which is dictated by the clinical demands of the application. In particular, we incorporate the standard dueling bandit algorithm with the dependence of arms that can be captured by some similarity function.

Contributions. This chapter study a novel class of dueling bandits problems – correlational dueling bandits with dependent arms and noisy comparison feedbacks. By adding the structure of correlation, we aim to achieve fast convergence for dueling bandits.

Concretely, the chapter proposes a novel Correlational Dueling algorithm as the incorporation of a Correlational Update subroutine and a Beat-the-Mean algorithm. It takes advantage of the correlations among different arms to update the whole active set of arms instead of only updating the two dueling arms. This achieves fast convergence to the (near) optimal decisions regardless of the large decision space.

Correlational Dueling is deployed as the first algorithmic approach to the control of spinal cord stimulation in clinical experiments. Correlational Dueling could find a group of optimal stimuli and help the paraplegic human patients to achieve full-weight standing.

The algorithm we developed in this paper has the potential to incorporate any of the previously mentioned MAB algorithms. Previous work has shown that dueling bandits algorithms enjoy state-of-the-art empirical performance using Sparring (Ailon, Z. Karnin, and Joachims, 2014) or RUCB (Zoghi, Whiteson, Remi Munos, et al., 2014).

6.2 Related Work

Correlated Bandits

In many applications, the set of candidate actions is very large (or even infinite). In such settings, dependence between the payoffs associated with different decisions must be modeled and exploited. Various methods of introducing dependence include bandits on trees (Kocsis and Szepesvári, 2006a), bandits with linear correlations (Dani, Hayes, and Kakade, 2008; Abernethy, Hazan, and Rakhlin, 2008; Abbasi-Yadkori, Pál, and Szepesvári, 2011) or Lipschitz continuous payoffs (Kleinberg, Slivkins, and Upfal, 2008; Sébastien Bubeck, Rémi Munos, et al., 2008), and Gaussian payoffs (Srinivas et al., 2010b). In this paper we pursue a Bayesian approach to bandits, where fine-grained assumptions on the regularity of the reward function can be imposed through proper choice of the prior distribution over the payoff function.

Dueling Bandits

Beyond the stochastic K-armed dueling bandits setting, other dueling bandit settings include multi-way preference feedback (Sui and J. Burdick, 2014b), continuous-armed convex dueling bandits (Yue and Joachims, 2009), contextual dueling bandits which also introduces the von Neumann winner solution concept (Dudik et al., 2015), sparse dueling bandits that focus on the Borda winner solution concept (Jamieson et al., 2015), Copeland dueling bandits that focus on the Copeland winner solution concept (Zoghi, Z. S. Karnin, et al., 2015), and adversarial dueling bandits (Gajane, Urvoy, and Clérot, 2015). It would be interesting to study how to extend the analysis to these other settings as well.

The dueling bandits problem can also be viewed as a special case of partial monitoring problems (Cesa-Bianchi, Lugosi, and Stoltz, 2006). In partial monitoring, the feedback received is assumed to be only indirectly related to the actual rewards. However, generic algorithms for partial monitoring problems are generally not competitive compared to algorithms specifically designed for the dueling bandits problem.

6.3 Problem Setup

The classical dueling bandit problem receives feedback in the form of a comparison between a pair of arms in each test. When the size of the decision set, K, is large, it is unavoidable to carry out a very large number of tests before the algorithm converges to its optimal solution. In some applications like the clinical example, each test is expensive and time consuming. The number of tests – time horizon of an algorithm – is often predetermined by clinical conditions. It is infeasible to apply the original dueling bandit algorithms to these applications due to the large decision space.

The procedure can be described as follows. There are K arms $\{b_1, \cdots, b_K\}$, and a total number of T tests to be performed. In the clinical spinal cord application, each test physically corresponds to a ~90-second stimulation period with a specific stimulus (arm) chosen from the K arms. T is determined before we run the algorithm. The K arms are correlated and $T \leq K$ in general.

Following the the original notation of the dueling bandit problem, for two arms b_i and b_j, where $i, j \in \{1, \cdots, K\}$, write the comparison factor as

$$\epsilon(b_i, b_j) = P(b_i \succ b_j) - 1/2,$$

where $P(b_i \succ b_j)$ is the probability that b_i dominates b_j and $\epsilon(b_i, b_j) \in [-1/2, 1/2]$ represents the priority between b_i and b_j. We define $b_i \succ b_j \Leftrightarrow \epsilon(b_i, b_j) > 0$. We use the notation $\epsilon_{i,j} \equiv \epsilon(b_i, b_j)$ for convenience. Note that $\epsilon(b_i, b_j) = -\epsilon(b_j, b_i)$ and $\epsilon(b_i, b_i) = 0$. We assume the distribution of reward for each arm is stationary so that all comparison factors converge in [-1/2,1/2]. We also assume $w.l.o.g.$ that the bandits are indexed in preferential order $b_1 \succ b_2 \succ \cdots \succ b_K$ so that there is one preferred arm.

The total reward is defined in terms of regret as in the classical bandit problem setting. In the online setting, let $b_{(t)}$ be the arm chosen at test t. We define total regret as follows:

$$R_T = \sum_{t=1}^{T} \epsilon(b_1, b_{(t)})$$

The total regret is zero, $R_T = 0$, if we constantly choose $b_{(t)} = b_1$ during the experiment. $R_T = \Theta(T)$ is linear $w.r.t.\, T$ if we constantly choose $b_{(t)} \in \{b_1, \cdots, b_K\}$.

The problem of correlational dueling bandits takes the correlations among arms into consideration. For any pair of arms b_i and b_j, we consider that the dependence between them can be captured by some similarity function $r_{ij} \in [0, 1]$, and it satisfies:

- $r_{ij} = r_{ji}$
- $r_{ij} = 0 \iff b_i$ and b_j are not correlated.
- $r_{ij} = 1 \iff b_i = b_j$.

We also inherit two properties of the comparison factors from the original dueling bandit problem:

Strong Stochastic Transitivity. For any triplet of arms $b_i \succ b_j \succ b_k$, we assume $\epsilon_{i,k} \geq \max\{\epsilon_{i,j}, \epsilon_{j,k}\}$.

Stochastic Triangle Inequality. For any triplet of arms $b_i \succ b_j \succ b_k$, we assume $\epsilon_{i,k} \leq \epsilon_{i,j} + \epsilon_{j,k}$. This can be viewed as a diminishing returns property.

In the application, we suppose that there exists an underlying utility function over the arms which we cannot observe directly. The observations are the noisy comparisons of utilities of different arms. These properties typically hold under the assumption.

Algorithm 13 Correlational Dueling

```
Input: $\mathcal{B}, T, (\kappa, \tau)$
Input: $c_\delta(n) = \sqrt{(1/n)log(1/\delta)}$
Run: [Parameters-Initialization]
Run: [Active-Elimination]
return $b^*$    // Optimal arm
```

Algorithm 14 Parameters-Initialization

```
$W_1 \leftarrow \mathcal{B}$    // set of active arms
$\ell \leftarrow 1$    // rounds
$\forall b \in W_\ell,\ n_b \leftarrow 0$    // comparisons
$\forall b \in W_\ell,\ w_b \leftarrow 0$    // priorities
$\forall b \in W_\ell,\ \hat{P}_b \equiv w_b/n_b$, or 1/2 if $n_b = 0$
$n^* \equiv min_{b \in W_\ell} n_b$
$c^* \equiv c_\delta(n^*)$, or 1 if $n^* = 0$    // confidence radius
$t \leftarrow 0$    // total number of iterations
return all new parameters
```

Algorithm 15 Active-Elimination

```
while $|W_\ell| > 1$ and $t \le T$ do
    select $b_i, b_j \in W_\ell$ at random
    compare selected arms (assume $b_i \succ b_j$)
    for all $b_k \in W_\ell$ do
        update $w_k, n_k$ by Correlational Update
    end for
    if $\min_{b' \in W_\ell} \hat{P}_{b'} + c^* \le \max_{b \in W_\ell} \hat{P}_b - c^*$ then
        $b' \leftarrow \arg\min_{b \in W_\ell} \hat{P}_b$
        $\forall b \in W_\ell$, delete comparisons with $b'$ from $w_b, n_b$
        $W_{\ell+1} \leftarrow W_\ell \backslash \{b'\}$    // update working set
        $\ell \leftarrow \ell + 1$    // new round
    end if
end while
return $b^* = \arg\max_{b \in W_\ell} \hat{P}_b$
```

Algorithm 16 Correlational Update

```
Input: $b_k, b_i \succ b_j$
$w_k \leftarrow w_k + \kappa(b_k; b_i, b_j)$
$n_k \leftarrow n_k + \tau(b_k; b_i, b_j)$
return $w_k, n_k$
```

6.4 Algorithm

Our algorithm, Correlational Dueling , is a correlational dueling bandits algorithm based on the Beat-the-Mean algorithm(Yue and Joachims, 2011). It uses observational feedback and the correlational structure to successively remove suboptimal arms, while keeping the optimal one(s) in the sample space. The inputs to Correlational Dueling are the set of arms $\mathcal{B}$, the total number of iterations T, and the correlational structure (κ, τ). $\kappa(b_k; b_i, b_j)$ and $\tau(b_k; b_i, b_j)$ control the weighted updates for b_k influenced by the comparison between b_i and b_j.

Parameters-Initialization (Algorithm 14) defines the set of active arms W_ℓ, whose size shrinks as more tests are completed. For each arm b, let n_b be the total number of comparisons between b and other arms, and let w_b be the total number of wins against all other arms. Let $\hat{P}_b$ be the empirical average of $P(b \succ b')$ for all b' in W_ℓ, and let $\hat{P}_{b,n}$ be the value of $\hat{P}_b$ after n comparisons between arm b and any other arms. Set the confidence interval of $P(b \succ b')$ as

$$\hat{C}_{b,n} = (\hat{P}_{b,n} - c_\delta(n), \hat{P}_{b,n} + c_\delta(n)),$$

where $c_\delta(n) = \sqrt{(1/n)log(1/\delta)}$, and δ is the confidence that $P(b \succ b')$ lies in $\hat{C}_{b,n}$.

The function $c_\delta(n)$ decreases as the number of comparisons n increases. By properly setting parameter δ, the optimal reward can be reached within the fixed time horizon.

Active-Elimination (Algorithm 15) is the key part of Correlational Dueling . For each pair of tests, two arms are randomly chosen from W_ℓ. The randomized selection method enjoys low-variance total regret in general. For each arm b, the values of w_b, n_b and $\hat{P}_b$ are updated, as is the corresponding confidence radius c^*. For any pair of arms b and b', one dominates the other if their confidence intervals do not overlap, and the less superior arm is eliminated from W_ℓ. The algorithm runs until the time horizon T is reached, or only one active arm remains.

Correlational Update (Algorithm 16) is the subroutine of *Active-Elimination* (Algorithm 15) which updates the weights of b_k by rules $\kappa(\cdot;\cdot,\cdot)$ and $\tau(\cdot;\cdot,\cdot)$.

In the classical dueling bandits setting, we assume arms are independent. For independent arms, if we have one comparison between b_i and b_j and gets $b_i \succ b_j$, we only update the weights for arm b_i and b_j:

$$w_i \leftarrow w_i + 1, n_i \leftarrow n_i + 1, \tag{6.1}$$

$$w_j \leftarrow w_j, n_j \leftarrow n_j + 1. \tag{6.2}$$

For large decision spaces (a lot of arms), even though some existing dueling bandits algorithms achieve optimal cumulative regret, the whole process is still extremely slow due to the independence among arms. When the arms are correlated and the correlation between any pair of arms b_i and b_j is measured properly by r_{ij}, we can update all active arms at each iteration.

In general, as shown in Algorithm 16, we update for every arm b_k after comparing arms b_i and b_j ($w.l.o.g.$ assume $b_i \succ b_j$) as follows:

$$w_k \leftarrow w_k + \kappa(b_k; b_i, b_j), \tag{6.3}$$

$$n_k \leftarrow n_k + \tau(b_k; b_i, b_j), \tag{6.4}$$

where $\kappa(\cdot;\cdot,\cdot)$ and $\tau(\cdot;\cdot,\cdot)$ represent the correlational structure. And it satisfies:

- $0 \leq \kappa(b_k; b_i, b_j) \leq \tau(b_k; b_i, b_j) \leq 1$;
- if $b_k = b_i$, $\kappa(b_k; b_i, b_j) = \tau(b_k; b_i, b_j) = 1$;
- if $b_k = b_j$, $\kappa(b_k; b_i, b_j) = 0$, $\tau(b_k; b_i, b_j) = 1$.

These updates are based on the assumption that $\kappa(\cdot;\cdot,\cdot)$ $\tau(\cdot;\cdot,\cdot)$ is an unbiased estimation of the dependent structure. The Correlational Update subroutine (Algorithm 16) can efficiently update all arms at each iteration. So Correlational Dueling enjoys fast convergence towards the near optimal arms.

Definition 1 *ε-optimal arm. If arm b satisfies $\epsilon(b_1, b) \leq \varepsilon$, then b is an ε-optimal arm.*

Proposition 1 *If $\exists\mu > 0$ such that $\tau(b_k; b_i, b_j) \geq \mu$ for every tuple $(b_i, b_j, b_k) \in \mathcal{B}^3$. then with high probability, the cumulative time to achieve purely ε-optimal arms $T(\varepsilon)$ is upper bounded by:*

$$T(\varepsilon) = \mathcal{O}\left(\frac{1}{\mu\varepsilon^2}\log\frac{1}{\delta}\right).$$

Proof. Proposition 1 holds based on the Theorem 1 of Yue and Joachims, 2011. After t iterations, since $\tau(b_k; b_i, b_j) \geq \mu$, we have $n^* \geq \mu t$. Then $c^* = c_\delta(n^*) = \sqrt{(1/n^*)log(1/\delta)} \leq \sqrt{(1/\mu t)log(1/\delta)}$. Notice c^* is a function of time step t.

For any arm b which is not ε-optimal (satisfies $\epsilon(b_1, b) > \varepsilon$), with probability $1 - \delta$, $\hat{P}_{b_1} - \hat{P}_b > \varepsilon C_\delta$ holds for some fixed concentration parameter C_δ. Suppose arm b has not been eliminated at iteration t. Then from elimination criterion Line 15 of Algorithm 15 we have $\varepsilon C_\delta < \hat{P}_{b_1} - \hat{P}_b < 2c^* \leq 2\sqrt{(1/\mu t)log(1/\delta)}$. The inequality breaks when $t \geq \frac{4}{\mu\varepsilon^2 C_\delta^2}\log\frac{1}{\delta} = \mathcal{O}\left(\frac{1}{\mu\varepsilon^2}\log\frac{1}{\delta}\right)$.

Notice, the iteration time $T(\varepsilon)$ in Propositions 1 does not depend on $|\mathcal{B}| = K$, which suggests the fast convergence of Correlational Dueling in large decision spaces.

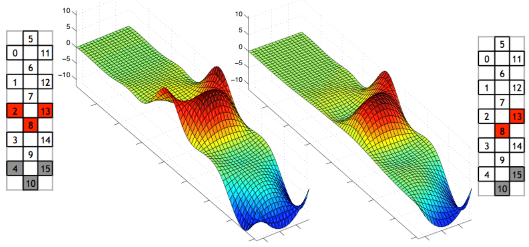


Figure 6.2: Capture the correlations between two different stimulating configurations.

For the epidural spinal stimulation application, we define the similarity of different configurations to be the correlation coefficient of electrical potential fields generated by the two different electrode stimulation configurations. We only update with the Correlational Update rule when $r(\cdot,\cdot) \geq 0$. The correlational property arises from analysis of electric fields applied by the array as shown in Figure 6.2.

The standard notion of correlation coefficient, $r_{XY} = E[XY - E[X]E[Y]]/\sqrt{Var[X]Var[Y]}$, is used in our experiments. However, one can use any measure as a basis for r_{XY} as long as $r_{XY} \in [0,1]$, $r_{XY} = 1$ when $X = Y$, and $r_{XY} = 0$ when X has an "irrelevant" relation to Y. The coefficient r can take negative values, but the algorithm doesn't use negative values to update Equations (5)(6).

For correlated arms, we perform an update for every arm k for which $r_{ik}, r_{jk} > 0$ as follows:

$$\kappa(b_k; b_i, b_j) \leftarrow \frac{\log r_{jk}}{\log r_{ik} + \log r_{jk}} \cdot \frac{r_{ik} + r_{jk}}{1 + r_{ij}}, \tag{6.5}$$

$$\tau(b_k; b_i, b_j) \leftarrow \frac{r_{ik} + r_{jk}}{1 + r_{ij}}. \tag{6.6}$$

Proposition 2 *If $\exists \mu > 0$ such that $r_{ij} \geq \mu$ for every pair $(b_i, b_j) \in \mathcal{B}^3$, then with high probability, the cumulative time to achieve purely ε-optimal arms $T(\varepsilon)$ satisfies:*

$$T(\varepsilon) = O\left(\frac{1}{\mu\varepsilon^2}\log\frac{1}{\delta}\right).$$

Proof. If $r_{ij} \geq \mu$ for every pair $(b_i, b_j) \in \mathcal{B}^3$, since $r_{ij} \leq 1$, $\tau(b_k; b_i, b_j) = \frac{r_{ik}+r_{jk}}{1+r_{ij}} \geq \frac{2\mu}{2} = \mu$ for every tuple (b_i, b_j, b_k). Substitute it into Proposition 1 and then Proposition 2 holds.

The Correlational Update subroutine above updates the dueling pair b_i, b_j in the same way as if they are independent since (5) and (6) will collapse to (1) and (2) for b_i and b_j. For extreme cases, if $b_i \succ b_j$ and arms b_k is very close to b_j, we have $r_{jk} \simeq 1$ and $r_{ik} \simeq r_{ij}$, the updating rules for arm b_k will be close to the updates of arm b_j. If b_k is far from both b_i and b_j, (5) and (6) guarantees that the update for b_k is very small since we acquire little information about b_k. Also, if b_i and b_j are less dependent (with smaller r_{ij}), we could acquire larger updates for the points in between.

A Bayesian optimization version can also be applied under this framework. We would assume the arms are sampled from a Gaussian process with Gaussian noise. In the experiments of this paper, we focus on the correlation coefficient as a compromise between clinical constraints and an explicit Bayesian updates.

6.5 Experiments

I first evaluated the algorithm on synthetic data. The algorithm was also applied to a real clinical application: online optimization for spinal cord stimulation therapy. The synthetic experiments seek to address the following questions: How does the algorithm compare against standard dueling bandit algorithms? How effective is it in terms of convergence? In particular, we compare the algorithm against Beat-the-Mean, RUCB, and Sparring algorithm with UCB1. These three algorithms are the representative dueling bandits algorithms designed for independent arms, which do not however respect the correlational arms.

Simulation Experiments

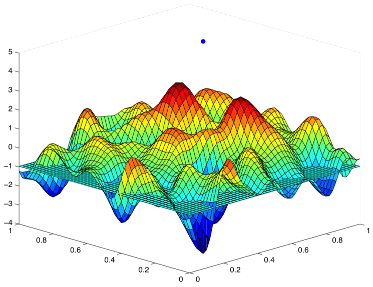


Figure 6.3: Mean function sampled from a Gaussian process.

These experiments validate the algorithm, and demonstrate its quick convergence when the arms are dependent. To generate correlated arms, a random functions is sampled from a zero-mean Gaussian Process with squared exponential kernel over the sample space $D = [0, 1] \times [0, 1]$, uniformly discretized into 50×50 points. This function defined the mean function for the 2500 arms, and $\sigma = 0.5$ was chosen as the standard deviation of the arms. One of the random functions is shown in Figure 6.3. The mean function is not necessarily convex or simple. Within each iteration, 2 points in the active set were sampled and their sampling values were compared to get the $\{0, 1\}$ feedback of the duel. The duel is completed over for $T = 100$ iterations for 10000 trials for each of the 4 comparing algorithms. I report a notion of *regret* as the stepwise regret instead of the cumulative regret. It converges to zero as iteration number goes to infinity for every no-regret algorithm. As seen in Figure 6.4, Correlational Dueling converges much faster than the other 3 algorithms since it takes the advantage of the dependent arms. The independent-armed dueling bandits algorithms require an exhaustive searching period which is significantly larger than the time horizon chosen here, before concentrating on the (near) optimal arms.

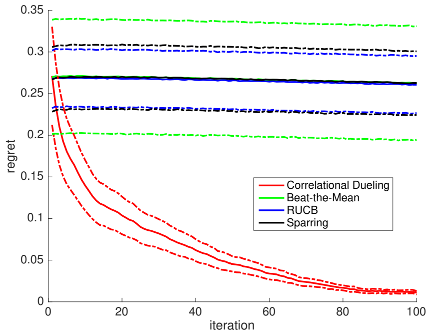


Figure 6.4: Regret versus iteration.

Human Experiments

A demonstration of the system for stand training under spinal cord stimulation with an spinal cord injury patient is shown in Figure 7.2. The subject practices standing under spinal stimulation using a stand frame for assistant in balance. The training processes largely follow the procedures in (Rejc, Angeli, and Harkema, 2015). Two trainers on the left and right protect and assist the subject. Within each experiment, a specific stimulating pattern (a combination of active electrode selections, the polarity of the actively selected electrodes, and the stimulation amplitude and frequency) is applied through the implanted electrode array and its controlling circuitry. An anonymous short video[1] shows the standing quality under different stimuli. The first part shows a low quality bipedal standing and the second part shows a better standing, both with electrical spinal cord stimulation. Different standings could look similar for the non-specialist.

The participants are under stable medical condition and have no painful musculoskeletal dysfunction that might interfere with stand training. They have no motor response present in leg muscles during transcranial magnetic stimulation, indicating that there are no strongly active neural pathways connecting cortex and lower limb muscles. No volitional control can be achieved during voluntary movement attempts in leg muscles as measured by EMG activity.

We also use clinical knowledge to restrict the decision space from around 4.3×10^7 to be on the order of $10^3 \sim 10^4$. It is still a very large decision space considering the number of trials, or arm pulls, are on the order of 10^2.

A total of 414 experimental comparisons were done with two patients under the Correlational Dueling algorithm. Each trial lasted for about 5 minutes. Within

[1] https://youtu.be/N9hK3ZagUSQ

each trial, one stimulating pattern was generated by the 16-channel electrode. The patterns were unchanged within each trial. For a fixed electrode configuration, the stimulation frequency and amplitude were modulated synergistically in order to find the best values for effective weight-bearing standing. We optimized the electrode patterns with Correlational Dueling and did an exhaustive search for stimulation frequency and amplitude over a narrow range.

Stimulation began while the patient was seated. Then the participant initiated the sit to stand transition by positioning his feet shoulder width apart and shifting his weight forward to begin loading the legs.

For the clinical experiments, we cannot create a direct plot for regrets since the ground truth optimal stimulation is unknown. In the experiments, we observed the convergence of Correlational Dueling , which is not possible for independent-armed dueling bandits algorithms. The set of (near) optimal configurations found by Correlational Dueling is shown in Figure 6.5. I compared the performance of Correlational Dueling to the optimal selections found heuristically for each patient by clinicians, which are shown in Figure 6.6. I found that the manual selection is a subset of the algorithm's selection, and there exist high performing configurations (e.g., the 2nd in Figure 6.5) found by the algorithm which are not in the manual selection. This shows that Correlational Dueling is performing no worse than specialized therapists.

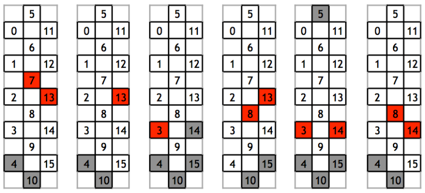


Figure 6.5: The set of (near) optimal configurations found by the algorithm for a specific patient (in decreasing order in terms of performances).

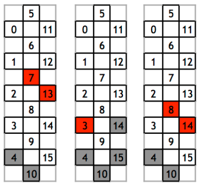


Figure 6.6: The set of (near) optimal configurations found by physician's manual pick for that specific patient (in decreasing order in terms of performances).

6.6 Conclusion and Discussion

The analysis and simulation demonstrate that Correlational Dueling indeed exhibits fast convergence properties comparing to independent-armed dueling bandits algorithm. This algorithm is demonstrated in clinical experiments for the control of spinal cord stimulation and showed that Correlational Dueling performs no worse than specialized physicians. This result provides an important step towards employing machine learning algorithms in many problems with a large volume of parameter selection and sequential decision making. These problems could be facilitated by our algorithm, which simultaneously delivers effective decisions and explores the decision space based on comparative feedback.

Correlational Update subroutine is easy to incorporate with Beat-the-Mean algorithm to achieve efficient Correlational Dueling . Although we developed Correlational Dueling specifically based on Beat-the-Mean, Correlational Update is a more general approach which has potential to incorporate with the existing dueling bandits algorithms. For instance, it can incorporate with RUCB to get a variant of RUCB for dependent arms by updating the wins w_{ij} with Correlational Update .

To my knowledge, this is the first applied algorithm towards spinal cord injury treatments. The algorithm could find a proper set of optimal stimulating configurations within the test time horizon. The approach achieved good performance in both simulations and human experiments. The paraplegic human patients could achieve full-weight standing under the stimulation provided by this algorithm.

Chapter 7

HUMAN EXPERIMENTS: UNDERSTANDING HUMAN STANDING VIA EMG MEASUREMENTS

Chapter 6 describes a procedure to optimize stimuli based on simple ranking of patient response. In order to get reliable measurements of performance beyond comparisons, we need to represent the quality of human standing quantitatively. Bipedal standing and walking are generally hard to achieve for both animals and robots. A human being, as an exception, has advanced mechanical structure and control system that is well-suited for bipedal movements. This chapter evaluates the standing behavior of paralyzed patients under spinal cord stimulation using Electromyographic recording (EMG). The quality of bipedal standing needs to be quantified to achieve an automatic approach.

7.1 Introduction

Bipedal standing and walking have been studied for both understanding of biological mechanisms and developing humanoid robots. Achieving stable standing is often considered to be easy for healthy adults and bipedal robots. But it is not trivial for young children when they are learning to stand. Our problem is motivated by the clinical research which aims to help paralyzed patients to stand up. Recent studies demonstrate the possibility of recovering motor function after severe SCI. Previous research (Harkema et al., 2011; Rejc, Angeli, and Harkema, 2015) has shown that electrical stimulation applied to the spinal cord via electrodes arrays implanted in the epidural space over the lumbosacral area (as shown in Figure 7.1) enables paralyzed patients to achieve full weight-bearing standing, improvements in stepping, and partial recovery of lost autonomic functions. These patients can only maintain standing under electrical spinal cord stimulation. We call this kind of standing "stimulated standing" to distinguish from the natural standing of healthy people. Compared to natural standing, artificial standing has several characteristics:
(1) Standing is mainly initiated by stimulation and sensory input instead of the patient's own will.
(2) The activity and strength of major muscle groups can be very different from the activity and strength under natural standing.
(3) Balance is much more difficult to achieve for standing. It requires fine tuning of

the stimuli and training by physicians.

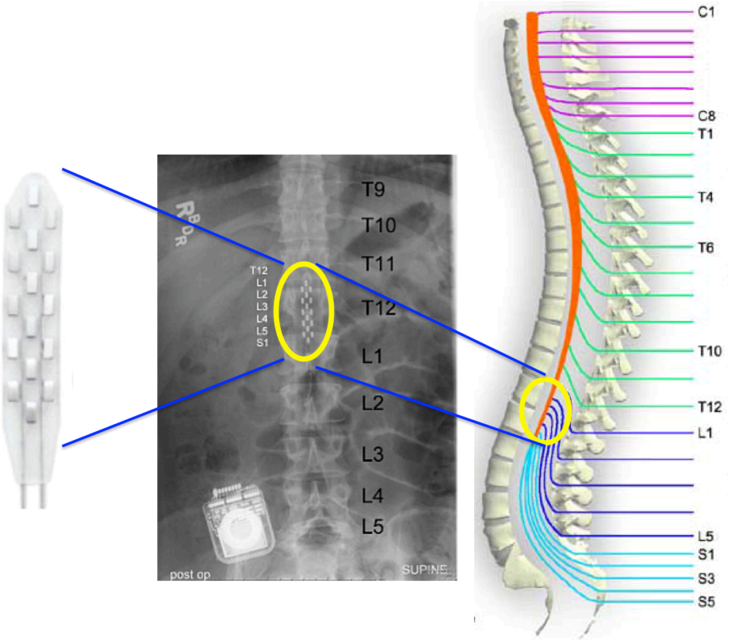


Figure 7.1: Spinal Cord Stimulation

As discussed in Chapter 6, a 16-electrode array is implanted for stimulation. The stimulation consists of electrical pulse trains applied to selected electrodes. The possible stimulus patterns (the choice of active electrodes and their polarity, the pulse amplitude and width, and the pulse train frequency) generate a huge space of parameters we can choose from. The choice of the parameters was optimized over time by a bandit algorithm in previous chapters. This chapter explores how to evaluate the stimulus response via 12-channel sEMG recording during the stimuli testing experiments.

To the best of my knowledge, this work is the first attempt to quantify standing performance of SCI patients using multi-channel sEMG. It is shown that even with very limited number of features and simple linear predicting model, the 12-channel EMG recording can provide accurate, fast and robust estimation for the quality of bipedal standing. Moreover, the total number of sEMG channels can be significantly reduced while keeping a high accuracy for estimation.

7.2 Related Work

Electrical stimulation can be used in multiple ways to enable or improve motor function in SCI. The data analyzed in this paper is relevant to the process of epidural spinal stimulation for human standing recovery. It has been shown by (Harkema et al., 2011; Rejc, Angeli, and Harkema, 2015) that when properly applied, this type of stimulation can enable paralyzed patients to achieve full weight-bearing standing. The results obtained in this intervention *are not* derived by direct stimulation of specific postural muscles, but by excitation of natural postural control circuits.

Functional Electrical Stimulation (FES), where electrical currents are applied to the intact peripheral motor nerves of paralyzed muscles to elicit muscle contractions, can provide significant levels of motor function (Peckham and Knutson, 2005). It is a widely used technique after SCI to enhance muscle strength and movements. EMG signals are used for on-line control of FES (Frigo et al., 2000). Posture shifting after spinal cord injury using functional neuro-muscular stimulation has been studied in computer simulation (Audu et al., 2011). Unlike FES, which has a direct mapping between neuro-muscular stimulation and muscle activity, the mapping between spinal cord stimulation and muscle activity is largely unknown. However, EMG activity is important to the use of both of these electrical stimulation modalities in SCI.

Traditional methods such as time-domain and frequency-domain analyses have been widely utilized in EMG pattern recognition (Phinyomark, Phukpattaranont, and Limsakul, 2012) Using EMG to predict movement, and control of robotic prostheses has been widely studied, as learning EMG control of a robotic hand (Bitzer and Van Der Smagt, 2006) or a wrist exoskeleton (Khokhar, Xiao, and Menon, 2010). EMG signals has also been used to control rehabilitation exoskeleton by paralyzed patients (Yin, Fan, and Xu, 2012), but not under the condition of spinal stimulation.

Biomechanical models are often built to simulate human standing and movement. They range from elegant inverse pendulum models (Winter, 2009), to more complicated musculo-skeletal models such as (Geyer and Herr, 2010), (Wang et al., 2012), and (Mordatch et al., 2013). Biomechanics and motor control of human movement are studied for the understanding of biological mechanisms, the developing of humanoid robots, and the virtual animation of human beings.

Healthy human plans for standing as a single task instead of the coordination of multiple tasks has been considered. The motor control mechanisms of severe SCI patients with spinal cord stimulation are largely unknown.

7.3 Methods

Data Acquisition and analysis.

EMG and ground reaction forces data were recorded at 2000 Hz using a custom-written acquisition software (National Instruments, Austin, TX). EMG activity of right (R) and left (L) gluteus maximus (GL), medial hamstring (MH), rectus femoris (RF), vastus lateralis (VL), tibialis anterior (TA), medial gastrocnemius (MG), and soleus (SOL) was recorded by means of bipolar surface electrodes with fixed inter-electrode distance. Bilateral EMG from the iliopsoas (IL) was recorded with fine-wire electrodes. Two surface electrodes were placed symmetrically lateral to the electrode array incision site over the paraspinal muscles in order to record the stimulation artefacts, which were used as indicators of the stimulation onset (time points when the stimulus pulses were applied). The time between stimulation onset and the EMG response onset was defined as the latency time of the evoked response. The amplitude of spinal cord evoked responses was quantified by peak to peak amplitude. The differences in amplitude were statistically evaluated by Student's paired t-test. To investigate the variability of the spinal cord evoked responses generated at different stimulation frequencies, the coefficient of variation (standard deviation / mean) was calculated over 20 ms after the onset of the spinal cord evoked responses (N = 20), which were selected within a representative portion of continuous (not rhythmic) EMG recording. Ground reaction forces were collected using a high-resolution pressure sensing mat (HR mat system, TEKSCAN, Boston, MA).

Human Experiments

A demonstration of the spinally stimulated human stand training experiments with an SCI subject is shown in Figure 7.2. The subject practices standing under spinal stimulation using a stand frame for assistance in achieving balance. A specific stimulating pattern is shown in the right part of Figure 7.2. Each stimulus is a combination of active electrode selections (red and gray sites), the polarity of the actively selected electrodes (red as anodes and gray as cathodes), and the stimulation amplitude and frequency. Within each experiment, a different stimulus is chosen by an active learning algorithm (Sui, Yue, and J. W. Burdick, 2017) and applied through the implanted electrode array and its controlling circuitry. Throughout the whole experiment, a variety of different stimulating patterns have been tested. The standing quality under stimulation ranged from independent stand to max-assisted standing as shown in Table 7.1. Multi-channel EMG signals were recorded and quantitative scores for standing were provided by physicians.

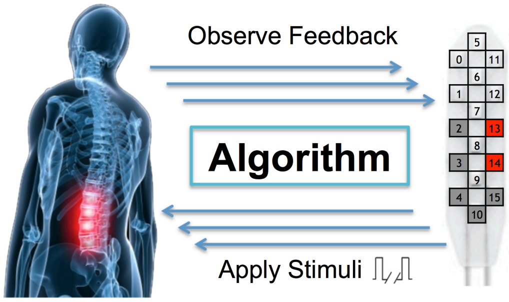


Figure 7.2: The Standing Experiment under spinal stimulation.

The participant is under stable medical condition and has no painful musculoskeletal dysfunction that might interfere with stand training. He has no motor response present in leg muscles during transcranial magnetic stimulation, indicating that there are no strongly active neural pathways connecting cortex and lower limb muscles. No volitional control can be achieved during voluntary movement attempts in leg muscles as measured by EMG activity.

A total of 109 experimental trials were done with the same patient. Each trial lasted around 5 minutes. Within each trial, one stimulating pattern was applied to the 16-channel electrode. The patterns were unchanged within each trial. For a fixed stimulating pattern, the stimulation frequency and amplitude were modulated synergistically in order to find the best values for effective weight-bearing standing. Different stimulating patterns are exploited along the trials in order to find the most effective ones. Specific electrode configuration adjustments were defined to seek improvements of different aspects of motor output. The guideline for parameter-tuning is outlined in our previous literature along with results of previous experiments performed on the same research participant. These constraints together with machine learning algorithms built on top of Sui and J. Burdick, 2014b and Sui, Gotovos, et al., 2015 were used to determine which electrode configurations, out of those potentially available ($\sim 4.3 \times 10^7$ combinations of electrodes), were to be examined in order to seek improvements of motor function for standing.

All EMG signals were sampled and recorded at 2000 Hz. Signals from right (R) and left (L) gluteus maximus (GL), medial hamstring (MH), vastus lateralis (VL), tibialis anterior (TA), medial gastrocnemius (MG), and soleus (SOL) were recorded by surface EMG electrodes. These six muscle groups are widely known to be activated during standing and walking motion.

The patient performed experimental and training sessions for standing using a custom designed standing frame composed of horizontal bars anterior and lateral to the individual. These bars were used for upper extremity support and balance assistance as needed. If the knees or hips flexed beyond a safe standing posture, external assistance was provided at the knees to promote extension, and at the hips to promote hip extension and anterior tilt. Facilitation was provided either manually by a trainer or by elastic bungee cords, which were attached between the two vertical bars of the standing apparatus. Mirrors were placed in front of the participant and laterally to him, in order to allow a better perception of the body position via visual feedback, conditioned on the lack of proprioceptive sensory feedback.

Stimulation began while the subject was seated. Then the participant initiated the sit to stand transition by positioning his feet shoulder width apart and shifting his weight forward to begin loading the legs. As shown in Figure 7.2, the participant used the horizontal bars of the standing apparatus during the transition phase to balance and to partially pull himself into a standing position. Trainers positioned at the pelvis and knees manually assisted the subject as needed during the sit to stand transition.

During sitting, little or negligible EMG activity of lower limb muscles was induced by epidural stimulation, showing that the weight-bearing related sensory information was needed to generate sufficient EMG patterns to effectively support full weight-bearing standing in spinally stimulated SCI.

Table 7.1 illustrates how did the clinicians quantify standing quality. Traditional measurements like center of pressure(COP) and center of mass(COM) cannot characterize the standing for paralyzed patients sufficiently. Typically, spinal cord injured patients do not stand and balance like normal subjects. Since there are no widely accepted quantitative measures for standing quality of paralyzed patients, we developed a 1-10 discrete scoring system. For scores 1 to 5, the standing is not independent but with less and less assistance by bungees or trainers. With limited experimental resources, the max/mod/min level of assistance is a robust measure we could get from experienced assisting therapists. For scores 6 to 10, the standing is independent and full-weight bearing. As the score increases, the standing is more natural, stable and lasts a longer time. After every trial, a score on general standing quality was assigned. Both video and multi-channel EMG were recorded during the experiments.

The research participants signed an informed consent for electrode implantation,

Table 7.1: The Scoring Criterions

Score	Descriptions
1-2	Assisted by bungees or trainers (max)
3-4	Assisted by bungees or trainers (mod)
5	Assisted by bungees or trainers (min)
6-7	Hip: Not assisted, back arched Knee: Not assisted, loss of extension during shifting
8-10	Hip: Not assisted, back straight Knee: Not assisted, extended during shifting

stimulation, and physiological monitoring studies approved by the University of Louisville and the University of California, Los Angeles Institutional Review Boards. The individuals described in this chapter have also given written informed consent to publish these case details.

Standing Model

Figure 7.3 shows the musculoskeletal model of the legs and trunk used in this work. It illustrates the locations of the uniarticular muscle tendon units (MTU) and the joints they actuate. The hip joint is extended by the gluteal muscles (GL) and flexed by the hip flexor muscles (HFL), while the knee joint is extended by the vastus lateralis (VL) and flexed by medial hamstring (MH). The tibialis anterior (TA) and the soleus (SOL) generate dorsiflexion and plantarflexion torques at the ankle, respectively. Medial gastrocnemius (MG) is also taken into consideration. The choice of muscles is based on previous clinical experiments and the planar model proposed in Geyer and Herr, 2010.

For the control of standing, this model could be redundant subject to the skeletal constraints. A subgroup of muscles {GL, VL, SOL, TA} might be enough to keep the standing posture stable. We will experimentally evaluate the redundancy of multi-channel EMG signals for predicting standing quality in the result section.

EMG Processing

Feature Selection. The 12-channel EMG signals of one experiment are shown in Figure 7.4 and Figure 7.5 for a single trial of the experiment. Traditional methods such as time-domain and frequency-domain analyses have been widely utilized in EMG pattern recognition (Phinyomark, Phukpattaranont, and Limsakul, 2012), and they have a good capability to track muscular changes. Other methods like Bayesian

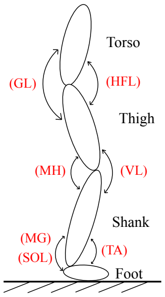


Figure 7.3: Musculo-skeletal Model.

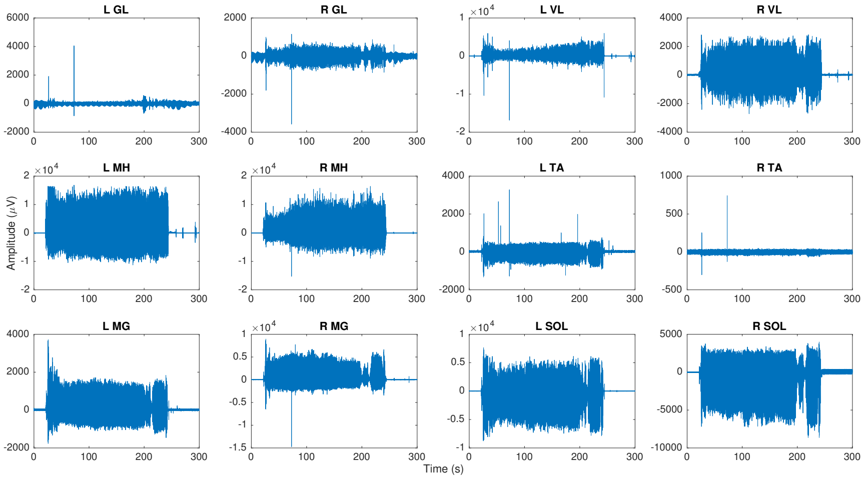


Figure 7.4: 12-channel EMG Signal. 'R GL' represents muscle GL on the right leg, etc. Amplitudes are not unified among figures for better representation.

estimations (Sanger, 2007) and linear filtering also achieves good estimations on muscle forces. We first consider simple and robust linear models with one estimator per channel. For each EMG channel, we calculate the mean power within 50 seconds at the early stage of standing and use it as the only feature for that channel. These 12 features were extracted in each trial and used in LDA and linear regression models for simple and robust predictions.

For the multi-class SVM features selection, we drew inspiration from previous works implementing machine learning techniques to predict forces applied at joints using EMG signals for exoskeleton control (Khokhar, Xiao, and Menon, 2010). A 4^{th} order Auto-Regressive(AR) model was fit to a 250 ms window of each EMG channel and the four coefficients (excluding the bias) were extracted as features. Thus, for 12-channels, a total of 48 features were extracted per observation. By performing

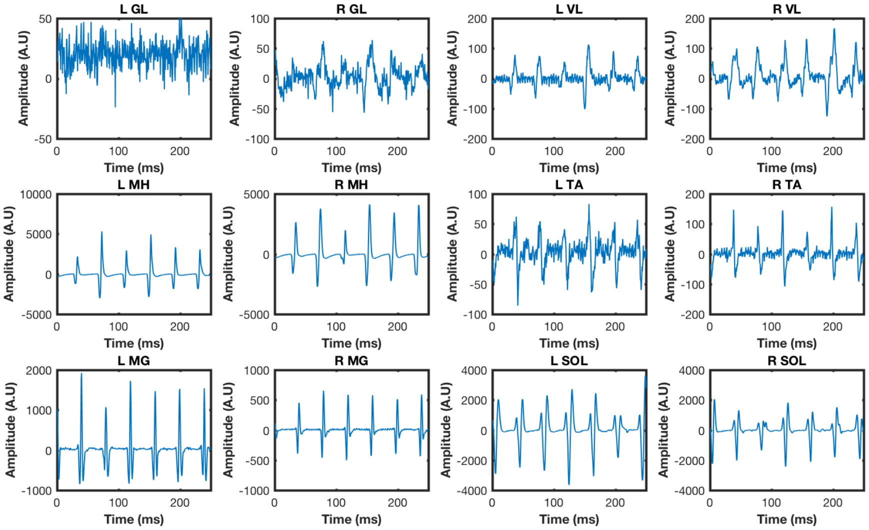


Figure 7.5: 12-channel EMG Signal. ’R GL’ represents muscle GL on the right leg, etc. Amplitudes are not unified among figures for better representation.

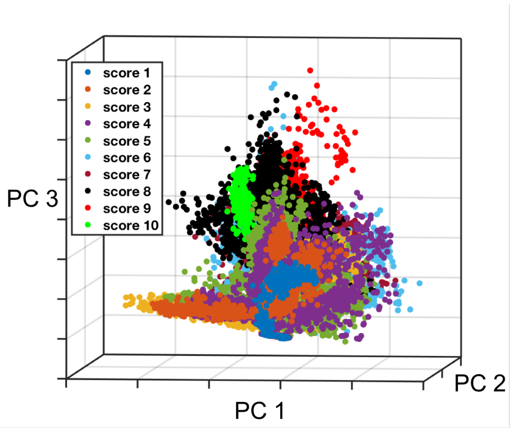


Figure 7.6: First 3 principal components of EMG dataset

10-fold cross validation on the optimum number of principal components we reduced the training set to the top 19 dimensions which capture 98% of the variance. Figure 7.6 shows the standing scores plotted against the first three principal components of the SVM training set. Even in 3-dimensions we see a high-degree of separability.

Model Selection.

As shown in Table 7.1, the data can be coarsely fit into 2 groups: good performances (not assisted, with score > 5) and bad performances (assisted, with score ≤ 5).

Linear discriminant analysis (LDA) is applied on the 2-class training data and predict whether a new group of EMG signals represents good or bad standing performance. A kernel-SVM model is trained for better accuracy. The SVM is trained to directly predict the standing quality score by translating the problem to a multi-class classification task with 10 classes (scores $1 - 10$). Each standing score corresponds to one class. A radial basis function with a scaling factor $\gamma = 0.79$ is used for the SVM kernel and a box constraint level of $C = 11$ was used to control the number of support vectors.

To show the robustness of EMG signals, the standing quality scores are estimated by directly applying linear regression on the scores from physicians v.s. 12-dimension power features.

7.4 Results

Estimating Standing Qualities

The original 12-channel surface EMG represents 6 muscle groups (GL, VL, MH, TA, MG, SOL) for both legs. For one of the high-performing standing experiments, the EMG waveform is shown in Fig. 7.5. 'R GL' represents right leg muscle GL, etc. The majority of muscles have strong and stationary EMG signals in this case.

First apply the LDA model with 12-dimension power features as input. The classification of good or bad performances yields an accuracy at 89.91%, which is a quite high rate conditioned on the limited number of features and simple LDA model. This classifier is good enough to be used in practice for a fast and robust decision on the quality of standing.

The kernel-SVM model yields 93.9% classification accuracy on the 10-class discrimination task upon 10-fold cross validation which confirms our belief that EMG signals are accurate predictors of bipedal standing. Moreover, using a more sophisticated model enables us to achieve higher classification accuracy then the linear model even with more classes. From the confusion plot in Fig. 7.7., one can see that most predictions lie within the range of the super diagonals indicating that it is highly unlikely for the SVM to mis-predict a score by a difference greater than 1. The percentages indicate the rate of true positives (white) and false negatives (red). The slots with rates less than 3% were omitted for succinctness. A standing score of 4 is the most often mis-predicted class due to its similarity with score 5 which can be attributed to the fact that these scores lie on the boundary between the mod and min level of assistance.

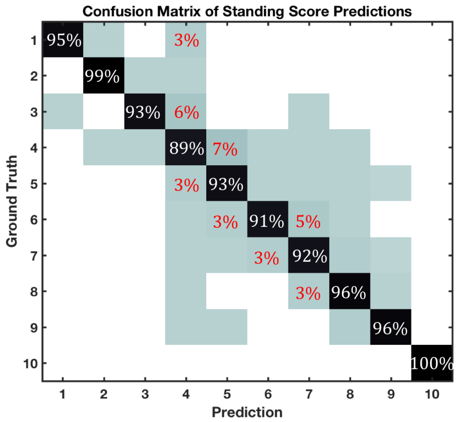


Figure 7.7: Confusion matrix of predictions made with SVM

To estimate the score for each experiment from EMG features, linear regression is also applied with 12-channel power features as inputs. As shown in Fig. 7.8, the x-axis represents true scores and y-axis measures the estimates by linear regression. The red line represents perfect match $y = x$. Each dot represents the true and estimated score of one experiment. The dots would be scattered close to the red line if the estimator is good. Within the 109 experiments, 57.8% of the estimates are within the region of true score ±1. And 93.6% of the estimates are within the region of true score ±2. Also 98.2% of the estimates are within the region of true score ±3. The standard deviation of estimating errors is 1.19, which is quite small comparing to the 1-10 scoring range.

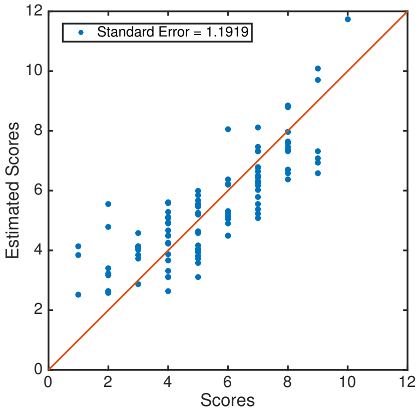


Figure 7.8: Regression on the Scores with 6 pairs (12-channel) EMG.

Reducing EMG Channels

Although more channels provide better estimation in general, in practice one may not have access to as many channels for all experiments. Also, fewer channels makes experiments efficient in time and budget. We investigate the possibility to reduce the number of EMG channels while keeping a high accuracy rate.

To choose the optimal $k \in \{1, 2, 3, 4, 5, 6\}$ sub-groups of muscles from the existing 6 muscle groups, we evaluate the classification/regression performance of the total $\binom{6}{k}$ muscle combinations. The optimal combination of muscle groups for each k is shown in Table 7.2. The single best muscle group for prediction is soleus (SOL).

Table 7.2: The Optimal Reducing Order of EMG Channels

Num. of Pairs	Optimal Combinations of EMG Channels
6	GL, VL, MH, TA, MG, SOL
5	VL, MH, TA, MG, SOL
4	VL, TA, MG, SOL
3	VL, MH, SOL
2	VL, SOL
1	SOL

Notice, this reduction process is different from principal component analysis (PCA) which reduces the feature space by picking the top independent components. Our approach aims at achieving good classification/regression by using fewer number of EMG channels. The chosen EMG channels may not be independent with each other.

Table 7.3 shows the optimal classification results with different number of muscle groups (channels) with 2-class LDA classification and 10-class SVM classification. For both models, the accuracy is slowly decreasing as the number of chosen muscle groups (k) decreases from $k = 6$ to $k = 2$. A quite high accuracy of 87.16% (for LDA, and 89.5% for SVM) is maintained even at $k = 2$. The muscle groups vastus lateralis (VL) and soleus (SOL) are the optimal combination for $k = 2$. One of them (SOL) is ankle flexor and the other (VL) is knee extensor as shown in 7.3. The accuracies drop significantly from $k = 2$ to $k = 1$. This makes sense since at lease 2 actuators are needed to control the 2 degrees of freedom. As a 2-class classification, LDA keeps a higher accuracy rate than SVM at $k = 1$.

Table 7.3: The Accuracies with Reducing Channels

Channels v.s.Accuracies	LDA(2-class)	SVM(10-class)
GL, VL, MH, TA, MG, SOL	89.91%	93.9%
VL, MH, TA, MG, SOL	88.91%	93.6%
VL, TA, MG, SOL	88.07%	93.0%
VL, MH, SOL	87.16%	92.7%
VL, SOL	87.16%	89.5%
SOL	80.73%	63.5%

7.5 Conclusions

Predictions

In this chapter, it is shown that multi-channel EMG recording can provide accurate, fast, and robust estimation for the quality of bipedal standing under spinal stimulation. I tested the effectiveness of spinal cord stimulation on a clinically sensory and motor complete participant. I introduced a musculo-skeletal model containing the major muscle groups that are involved in stable standing, and used it to explain the feasibility of reducing EMG channels.

The potential for reducing EMG channels was experimentally evaluated, which confirms that the 12-channel EMG signals are highly redundant for predicting standing quality. I showed the optimal combinations provide high scoring accuracy, and this performance can be maintained at a high level even with few channels. This fact contradicts to my initial assumption that multiple muscle group coordination is essential, with no dominant component to be expected. However, better estimation for the standing quality requires recording from larger number of muscle groups.

There are multiple ways to improve the accuracy of the predictions. This chapter already demonstrated that using more elaborate features with an SVM achieves high prediction accuracy even for the multi-class problem. Including even more features from raw EMG signals and using finer picked/tuned models is one venue for improvement. As mentioned in Section 7.1, the activity and strength of major muscle groups in stimulated SCI patients can be very different from the activity and strength under natural standing. Usually, it contains an early response strongly modulated by the electrical stimulation and a late response which is more like the EMG patterns from healthy subjects. Separating these two stages for feature extraction should also improve the predictability. Under a budget constraint on the number of channels, asymmetric placement of EMG sensors on left and right legs could also improve the accuracy assuming the stimulation effects equally for the two sides. We could also consider adding more physical measurements for prediction conditioned on the experimental environments.

In general, the scoring, proposed in this chapter estimators can provide reliable scores on the quality of patient standing when experienced physicians are not available during experiments.

Sensor Placement Efficiency

I have shown the optimal EMG channel combination subject to a budget constraint on the number of channels. What if we have more EMG sensors to place? Previous research (Gartman, 2008) suggested that the prime muscle targets should be a set of 8×2 muscles supporting 42% of the standing postures. Coactivation of an extra 4×2 muscles increased the percentage of feasible postures to 71%. We can sample from this larger space and it may reduce to a better group of 12 channels than the current 12 channels.

Combining with Exoskeleton

A large group of people including spinal cord injured patients often need rehabilitation robotic systems to provide functional gait therapies or assist their standing and moving. Current assistive standing systems rarely takes feedback other that direct force measurements from users. The automatic approach to quantify the quality of bipedal standing by EMG could provide estimation of standing quality to the assistive systems for better standing control. More efficient gait therapies and movement control could be achieved by incorporating EMG measures into the rehabilitation robotic systems.

Chapter 8

CONCLUSION AND FUTURE WORK

8.1 Conclusions

This dissertation develops a series of new algorithms for optimization in uncertain environments: Rank-Comparison, SelfSparring-IW, SelfSparring, Correlational Dueling , Local-UCB, and SafeOpt. The theoretical guarantees on the performances for Rank-Comparison, SelfSparring, and SafeOpt are provided.

The algorithms have successfully managed the optimization process under uncertainties. For spinal cord injury therapy, the online learning of spinal cord stimulation has not previously been successfully executed by any algorithms with human patients. These results represent a substantial step toward both autonomously adaptive neural stimulation for spinal cord injury and lower-cost EES-based therapies for patients. These techniques may also be applicable to other multi-electrode neural stimulation problems, such as deep brain stimulation and functional electrical stimulation, etc.

Optimization with relative feedback

Chapter 3 and Chapter 4 of this dissertation develop the simple and efficient SelfSparring and RankComparison algorithms for optimization with relative feedback. These chapters also develop theoretical, high probability bounds on the regret of the algorithms.

Safe exploration for optimization

Chapter 5 describes the safe optimization approach with Gaussian processes. We investigated the novel problem of trading exploration and exploitation for function optimization under safety constraints. In particular, we proposed SafeOpt, an efficient algorithm that balances the tradeoff between expanding, exploring and optimizing over the reachable safe region. We prove strong theoretical performance guarantees for SafeOpt, bounding its sample complexity to achieve an ε-optimal solution while guaranteeing safety with high probability.

In simulation studies, including both synthetic and real data, SelfSparring and SafeOpt algorithms attained state-of-the-art performances.

Therapy for spinal cord injuries

Chapter 6 studies the effectiveness of Correlational Dueling with human experiments. It is the first applied algorithm on spinal cord injury treatments. The algorithm could find a proper set of optimal stimulating configurations within the test time horizon. This chapter shows good performances in both simulations and human experiments. The paralyzed human patients could achieve full-weight standing under the stimulation provided by the algorithm.

Quantifying patient standing

Chapter 7 shows that multi-channel sEMG recording can provide accurate, fast and robust estimation for the quality of bipedal standing. We tested the effectiveness of spinal cord stimulation on a clinically sensory and motor complete participant. It showed that the patient was able to stand over-ground bearing full body-weight without any external assistance, using their hands to assist balance. This thesis demonstrates a musculo-skeletal model containing the major muscle groups that are involved in stable standing and uses it to explain the feasibility of reducing EMG channels.

8.2 Future Works

Towards general Safe Optimization

This thesis considers the safe optimization problem within which the safety function and reward function are the same. A more general framework would treat safety constraint and reward function separately. It is also interesting to consider multiple safety constraints simultaneously.

Towards more complex stimulation

The optimization over a high dimensional input space of stimulating configurations is studied in this thesis. However, an even larger input space is waiting to be explored if we consider stimulation in temporal space. A sequence of combined configurations may enhance the performance more than single ones as we've already seen in some pilot clinical experiments. Stimulation for multiple tasks is also an important issue, as we are extending the control for standing to stepping and more complex behaviors.

Quantify the quality of human behaviors

For gait in particular, these techniques generally require extensive manual annotation of video recordings, which would necessitate a long feedback loop. If a rough but sufficient analysis could be automated (e.g., pattern recognition on the raw motion capture trajectories, or on the EMG activity), these activities could be used in real time. We are looking forward to building an automatic grading system based on the classification and regression of multiple measures of human behaviors.

Towards a full autonomous system

Another improvement in clinical experiments would be to make the entire system fully autonomous. All experiments done by this thesis were performed with human researchers controlling the stimulator and the recording system, while the algorithm performed an executive or directing role. This architecture has a number of advantages, among them that the human experimenter provides a fail-safe with respect to data acquisition (e.g., if an element of the data processing fails, the observations of the human experimenter can often be used to reconstruct the missing information). Creating an integrated system in which the algorithm is controlling the data acquisition in (nearly) real time would allow a substantial acceleration of the testing process, and would constitute a very substantial step toward the goal of autonomy, a crucial requirement for a home-use or implantable device. For real time data acquisition and control, the electrode array should be controlled through software interface instead of manual input. A safety system would have to be created, that would allow the user to veto stimuli and terminate the delivery of stimuli which were distressing to the patient.

BIBLIOGRAPHY

Abbasi-Yadkori, Yasin, Dávid Pál, and Csaba Szepesvári (2011). “Improved Algorithms for Linear Stochastic Bandits.” In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 2312–2320.

Abernethy, Jacob, Elad Hazan, and Alexander Rakhlin (2008). “Competing in the Dark: An Efficient Algorithm for Bandit Linear Optimization”. In: *Proceedings of the 21st Annual Conference on Learning Theory (COLT)*, pp. 263–274.

Agrawal, Shipra and Navin Goyal (2012). “Analysis of Thompson Sampling for the Multi-armed Bandit Problem”. In: *Conference on Learning Theory (COLT)*.

Ailon, Nir, Zohar Karnin, and Thorsten Joachims (2014). “Reducing Dueling Bandits to Cardinal Bandits”. In: *International Conference on Machine Learning (ICML)*.

Anderson, Kim D (2004). “Targeting recovery: priorities of the spinal cord-injured population”. In: *Journal of neurotrauma* 21.10, pp. 1371–1383.

Audibert, Jean-Yves, Sebastien Bubeck, and Remi Munos (2010). “Best Arm Identification in Multi-Armed Bandits”. In: *Conference on Learning Theory (COLT)*.

Audu, Musa L et al. (2011). “Posture shifting after spinal cord injury using functional neuromuscular stimulation—a computer simulation study”. In: *Journal of biomechanics* 44.9, pp. 1639–1645.

Auer, Peter (2002). “Using Confidence Bounds for Exploitation-Exploration Tradeoffs”. In: *Journal of Machine Learning Research (JMLR)*.

Auer, Peter, Nicolo Cesa-Bianchi, and Paul Fischer (2002). “Finite-time analysis of the multiarmed bandit problem”. In: *Machine learning* 47.2-3, pp. 235–256.

Auer, Peter, Nicolo Cesa-Bianchi, Yoav Freund, et al. (2002). “The nonstochastic multiarmed bandit problem”. In: *SIAM Journal on Computing* 32.1, pp. 48–77.

Bareyre, Florence M et al. (2004). “The injured spinal cord spontaneously forms a new intraspinal circuit in adult rats”. In: *Nature neuroscience* 7.3, pp. 269–277.

Bitzer, Sebastian and Patrick Van Der Smagt (2006). “Learning EMG control of a robotic hand: towards active prostheses”. In: *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE, pp. 2819–2823.

Brand, Rubia van den et al. (2012). “Restoring voluntary control of locomotion after paralyzing spinal cord injury”. In: *science* 336.6085, pp. 1182–1185.

Brochu, Eric, Vlad M. Cora, and Nando de Freitas (2010). “A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning”. In: *arXiv:1012.2599*.

Brost, Brian et al. (2016). “Multi-dueling bandits and their application to online ranker evaluation”. In: *ACM Conference on Information and Knowledge Management*.

Bryan, Brent et al. (2005). “Active Learning For Identifying Function Threshold Boundaries”. In: *Neural Information Processing Systems (NIPS)*.

Bubeck, Sébastien and Nicolo Cesa-Bianchi (2012). “Regret analysis of stochastic and nonstochastic multi-armed bandit problems”. In: *Foundations and Trends in Machine Learning* 5, pp. 1–122.

Bubeck, Sébastien, Rémi Munos, et al. (2008). “Online Optimization in X-Armed Bandits”. In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 201–208.

Bubeck, Sébastien, Vianney Perchet, and Philippe Rigollet (2013). “Bounded regret in stochastic multi-armed bandits.” In: *COLT*, pp. 122–134.

Cai, Lance L et al. (2006). “Implications of assist-as-needed robotic step training after a complete spinal cord injury on intrinsic strategies of motor learning”. In: *The Journal of neuroscience* 26.41, pp. 10564–10568.

Capogrosso, Marco et al. (2016). “A brain–spine interface alleviating gait deficits after spinal cord injury in primates”. In: *Nature* 539.7628, pp. 284–288.

Cesa-Bianchi, Nicolo, Gábor Lugosi, and Gilles Stoltz (2006). “Regret minimization under partial monitoring”. In: *Mathematics of Operations Research* 31.3, pp. 562–580.

Chapelle, Olivier, Thorsten Joachims, et al. (2012). “Large-scale validation and analysis of interleaved search evaluation”. In: *ACM Transactions on Information Systems (TOIS)* 30.1, 6:1–6:41.

Chapelle, Olivier and Lihong Li (2011). “An empirical evaluation of thompson sampling”. In: *Advances in Neural Information Processing Systems (NIPS)*.

Courtine, Gregoire et al. (2008). “Recovery of supraspinal control of stepping via indirect propriospinal relay connections after spinal cord injury”. In: *Nature medicine* 14.1, pp. 69–74.

Courtine, Grégoire et al. (2005). “Performance of locomotion and foot grasping following a unilateral thoracic corticospinal tract lesion in monkeys (Macaca mulatta)”. In: *Brain* 128.10, pp. 2338–2358.

Cox, Dennis D and Susan John (1997). “SDO: A statistical method for global optimization”. In: *Multidisciplinary design optimization: state of the art*, pp. 315–329.

Crown, Eric D and James W Grau (2001). “Preserving and restoring behavioral potential within the spinal cord using an instrumental training paradigm”. In: *Journal of Neurophysiology* 86.2, pp. 845–855.

Dani, Varsha, Thomas P. Hayes, and Sham M. Kakade (2008). “Stochastic linear optimization under bandit feedback”. In: *Proceedings of the 21st Annual Conference on Learning Theory (COLT)*, pp. 355–366.

Desautels, Thomas Anthony (2014). “Spinal cord injury therapy through active learning”. PhD thesis. California Institute of Technology.

Desautels, Thomas, Andreas Krause, and Joel Burdick (2012). “Parallelizing Exploration-Exploitation Tradeoffs with Gaussian Process Bandit Optimization”. In: *International Conference on Machine Learning (ICML)*.

Dimitrijevic, Milan R, Yuri Gerasimenko, and Michaela M Pinter (1998). “Evidence for a spinal central pattern generator in humansa”. In: *Annals of the New York Academy of Sciences* 860.1, pp. 360–376.

Dudik, Miroslav et al. (2015). “Contextual Dueling Bandits”. In: *Conference on Learning Theory (COLT)*.

Edgerton, V Reggie et al. (2006). “Rehabilitative therapies after spinal cord injury”. In: *Journal of neurotrauma* 23.3-4, pp. 560–570.

Emken, Jeremy L et al. (2008). “Feasibility of manual teach-and-replay and continuous impedance shaping for robotic locomotor training following spinal cord injury”. In: *IEEE Transactions on Biomedical Engineering* 55.1, pp. 322–334.

Even-Dar, Eyal, Shie Mannor, and Yishay Mansour (2002). “PAC bounds for multi-armed bandit and Markov decision processes”. In: *Computational Learning Theory*. Springer, pp. 255–270.

Fawcett, JW et al. (2007). “Guidelines for the conduct of clinical trials for spinal cord injury as developed by the ICCP panel: spontaneous recovery after spinal cord injury and statistical power needed for therapeutic clinical trials”. In: *Spinal cord* 45.3, pp. 190–205.

Friedman, Eric and Scott Shenker (1998). *Learning and implementation on the Internet*. Tech. rep. Working Papers, Department of Economics, Rutgers, The State University of New Jersey.

Frigo, C et al. (2000). “EMG signals detection and processing for on-line control of functional electrical stimulation”. In: *Journal of Electromyography and Kinesiology* 10.5, pp. 351–360.

Fruitet, Joan, Alexandra Carpentier, Maureen Clerc, et al. (2012). “Bandit Algorithms boost Brain Computer Interfaces for motor-task selection of a brain-controlled button”. In: *Advances in Neural Information Processing Systems*, pp. 449–457.

Fruitet, Joan, Alexandra Carpentier, Rémi Munos, et al. (2013). “Automatic motor task selection via a bandit algorithm for a brain-controlled button”. In: *Journal of neural engineering* 10.1, p. 016012.

Gajane, Pratik, Tanguy Urvoy, and Fabrice Clérot (2015). “A Relative Exponential Weighing Algorithm for Adversarial Utility-based Dueling Bandits”. In: *International Conference on Machine Learning (ICML).*

Garcia, Javier and Fernando Fernandez (2012). “Safe Exploration of State and Action Spaces in Reinforcement Learning”. In: *JMLR*.

Gartman, Steven J (2008). “Selection of optimal muscle set for 16-channel standing neuroprosthesis”. In: *Journal of rehabilitation research and development* 45.7, p. 1007.

Gerasimenko, Yury, Roland R Roy, and V Reggie Edgerton (2008). “Epidural stimulation: comparison of the spinal circuits that generate and control locomotion in rats, cats and humans”. In: *Experimental neurology* 209.2, pp. 417–425.

Geyer, Hartmut and Hugh Herr (2010). “A muscle-reflex model that encodes principles of legged mechanics produces human walking dynamics and muscle activities”. In: *Neural Systems and Rehabilitation Engineering, IEEE Transactions on* 18.3, pp. 263–273.

Gillula, Jeremy and Claire Tomlin (2011). “Guaranteed safe online learning of a bounded system”. In: *IROS*.

Gittins, John, Kevin Glazebrook, and Richard Weber (2011). *Multi-armed bandit allocation indices*. John Wiley & Sons.

Gonzalez, Javier et al. (2016). “Bayesian Optimisation with Pairwise Preferential Returns”. In: *NIPS Workshop on Bayesian Optimization*.

Gotovos, Alkis et al. (2013). “Active Learning for Level Set Estimation”. In: *International Joint Conference on Artificial Intelligence (IJCAI).*

Gürel, Tayfun and Carsten Mehring (2012). “Unsupervised adaptation of brain machine interface decoders”. In: *arXiv preprint arXiv:1206.3666*.

Hans, Alexander, Daniel Schneegaßand Anton Schäfer, and Steffen Udluft (2008). “Safe exploration for reinforcement learning”. In: *ESANN*.

Harkema, Susan et al. (2011). “Effect of epidural stimulation of the lumbosacral spinal cord on voluntary movement, standing, and assisted stepping after motor complete paraplegia: a case study”. In: *The Lancet* 377.9781, pp. 1938–1947.

Hennig, Philipp and Christian J Schuler (2012). “Entropy search for information-efficient global optimization”. In: *Journal of Machine Learning Research* 13.Jun, pp. 1809–1837.

Herman, R et al. (2002). “Spinal cord stimulation facilitates functional walking in a chronic, incomplete spinal cord injured”. In: *Spinal cord* 40.2, pp. 65–68.

Jamieson, Kevin et al. (2015). “Sparse Dueling Bandits”. In: *International Conference on Artificial Intelligence and Statistics (AISTATS).*

Jones, Donald R., Matthias Schonlau, and William J. Welch (1998). “Efficient Global Optimization of Expensive Black-Box Functions”. In: *Journal of Global Optimization*.

Kaufmann, Emilie, Nathaniel Korda, and Rémi Munos (2012). “Thompson sampling: An asymptotically optimal finite-time analysis”. In: *Algorithmic Learning Theory (ALT)*.

Khokhar, Zeeshan O, Zhen G Xiao, and Carlo Menon (2010). “Surface EMG pattern recognition for real-time control of a wrist exoskeleton”. In: *Biomedical engineering online* 9.1, p. 41.

Kleinberg, Robert, Aleksandrs Slivkins, and Eli Upfal (2008). “Multi-armed bandits in metric spaces”. In: *ACM Symposium on Theory of Computing (STOC)*. Association for Computing Machinery, Inc., pp. 681–690.

Klose, K John et al. (1997). “Evaluation of a training program for persons with SCI paraplegia using the Parastep® 1 ambulation system: part 1. Ambulation performance and anthropometric measures”. In: *Archives of physical medicine and rehabilitation* 78.8, pp. 789–793.

Kocsis, Levente and Csaba Szepesvári (2006a). “Bandit Based Monte-Carlo Planning”. In: *Machine Learning: ECML*, pp. 282–293.

– (2006b). “Bandit based monte-carlo planning”. In: *European conference on machine learning*. Springer, pp. 282–293.

Komiyama, Junpei et al. (2015). “Regret Lower Bound and Optimal Algorithm in Dueling Bandit Problem.” In: *COLT*, pp. 1141–1154.

Lai, Tze Leung and Herbert Robbins (1985). “Asymptotically efficient adaptive allocation rules”. In: *Advances in applied mathematics* 6.1, pp. 4–22.

Liberson, WT et al. (1961). “Functional electrotherapy: stimulation of the peroneal nerve synchronized with the swing phase of the gait of hemiplegic patients.” In: *Archives of physical medicine and rehabilitation* 42, pp. 101–105.

Liu, Tie-Yan et al. (2007). “Letor: Benchmark dataset for research on learning to rank for information retrieval”. In: *SIGIR 2007 workshop on learning to rank for information retrieval*, pp. 3–10.

Liu, Zhao (2016). “Electromyographic Signal Processing With Application To Spinal Cord Injury”. PhD thesis. California Institute of Technology.

Minassian, Karen et al. (2007). “Human lumbar cord circuitries can be activated by extrinsic tonic input to generate locomotor-like activity”. In: *Human movement science* 26.2, pp. 275–295.

Mockus, Jonas (1989). *Bayesian Approach to Global Optimization*. Kluwer Academic Publishers.

Moldovan, Teodor and Pieter Abbeel (2012). “Safe Exploration in Markov Decision Processes”. In: *ICML*.

Mordatch, Igor et al. (2013). “Animating human lower limbs using contact-invariant optimization”. In: *ACM Transactions on Graphics (TOG)* 32.6, p. 203.

Peckham, P Hunter and Jayme S Knutson (2005). “Functional Electrical Stimulation for Neuromuscular Applications*”. In: *Annu. Rev. Biomed. Eng.* 7, pp. 327–360.

Phinyomark, Angkoon, Pornchai Phukpattaranont, and Chusak Limsakul (2012). “Feature reduction and selection for EMG signal classification”. In: *Expert Systems with Applications* 39.8, pp. 7420–7431.

Rasmussen, Carl Edward and Christopher K. I. Williams (2006). *Gaussian Processes for Machine Learning*. MIT Press.

Rejc, Enrico, Claudia Angeli, and Susan Harkema (2015). “Effects of lumbosacral spinal cord epidural stimulation for standing after chronic complete paralysis in humans”. In: *PloS one* 10.7, e0133998.

Robbins, Herbert (1952). “Some aspects of the sequential design of experiments”. In: *Bulletin of the American Mathematical Society*.

Russo, Daniel and Benjamin Van Roy (2014). “Learning to optimize via posterior sampling”. In: *Mathematics of Operations Research* 39.4, pp. 1221–1243.

Ryzhov, Ilya O, Warren B Powell, and Peter I Frazier (2012). “The knowledge gradient algorithm for a general class of online learning problems”. In: *Operations Research* 60.1, pp. 180–195.

Sanger, Terence D (2007). “Bayesian filtering of myoelectric signals”. In: *Journal of neurophysiology* 97.2, pp. 1839–1845.

Santaniello, Sabato et al. (2011). “Closed-loop control of deep brain stimulation: a simulation study”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 19.1, pp. 15–24.

Schölkopf, Bernhard and Alex J. Smola (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. The MIT Press.

Schuth, Anne et al. (2014). “Multileaved comparisons for fast online evaluation”. In: *ACM Conference on Conference on Information and Knowledge Management*.

Shealy, C Norman, J Thomas Mortimer, and James B Reswick (1967). “Electrical inhibition of pain by stimulation of the dorsal columns: preliminary clinical report.” In: *Anesthesia & Analgesia* 46.4, pp. 489–491.

Shealy, C Norman, Norman Taslitz, et al. (1967). “electrical inhibition of pain: experimental evaluation*.” In: *Anesthesia & Analgesia* 46.3, pp. 299–305.

Srinivas, Niranjan et al. (2010a). “Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design”. In: *International Conference on Machine Learning (ICML).*

– (2010b). “Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design”. In: *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pp. 1015–1022.

Sui, Yanan and Joel Burdick (2014a). “Bandit Problem With Subgroup Rank Feedback”. In: *arXiv preprint arXiv:0971348*.

– (2014b). “Clinical online recommendation with subgroup rank feedback”. In: *ACM Conference on Recommender Systems (RecSys)*.

Sui, Yanan, Alkis Gotovos, et al. (2015). “Safe exploration for optimization with Gaussian processes”. In: *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pp. 997–1005.

Sui, Yanan, Yisong Yue, and Joel W Burdick (2017). “Correlational Dueling Bandits with Application to Clinical Treatment in Large Decision Space”. In: *International Joint Conference on Artificial Intelligence (IJCAI)*.

Syrgkanis, Vasilis et al. (2015). “Fast convergence of regularized learning in games”. In: *Advances in Neural Information Processing Systems*.

Thompson, William R (1933). “On the likelihood that one unknown probability exceeds another in view of the evidence of two samples”. In: *Biometrika* 25.3/4, pp. 285–294.

Thuret, Sandrine, Lawrence DF Moon, and Fred H Gage (2006). “Therapeutic interventions after spinal cord injury”. In: *Nature Reviews Neuroscience* 7.8, pp. 628–643.

Urvoy, Tanguy et al. (2013). “Generic Exploration and K-armed Voting Bandits”. In: *International Conference on Machine Learning (ICML)*.

Vidaurre, Carmen et al. (2011). “Co-adaptive calibration to improve BCI efficiency”. In: *Journal of neural engineering* 8.2, p. 025009.

Wang, Jack M et al. (2012). “Optimizing locomotion controllers using biologically-based actuators and objectives”. In: *ACM transactions on graphics* 31.4.

Winter, David A (2009). *Biomechanics and motor control of human movement*. John Wiley & Sons.

Yin, Yue H, Yuan J Fan, and Li D Xu (2012). “EMG and EPP-integrated human–machine interface between the paralyzed and rehabilitation exoskeleton”. In: *Information Technology in Biomedicine, IEEE Transactions on* 16.4, pp. 542–549.

Yue, Yisong and Thorsten Joachims (2009). “Interactively optimizing information retrieval systems as a dueling bandits problem”. In: *International Conference on Machine Learning (ICML)*.

– (2011). “Beat the mean bandit”. In: *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 241–248.

Yue, Yisong et al. (2009). “The k-armed dueling bandits problem”. In: *Conference on Learning Theory (COLT)*.

– (2012). “The K-armed Dueling Bandits Problem”. In: *Journal of Computer and System Sciences* 78.5, pp. 1538–1556.

Zoghi, Masrour, Zohar S Karnin, et al. (2015). “Copeland Dueling Bandits”. In: *Advances in Neural Information Processing Systems*, pp. 307–315.

Zoghi, Masrour, Shimon Whiteson, Remi Munos, et al. (2014). “Relative Upper Confidence Bound for the K-armed Dueling Bandit Problem”. In: *International Conference on Machine Learning (ICML)*.

Zoghi, Masrour, Shimon Whiteson, and Maarten de Rijke (2015). “MergeRUCB: A method for large-scale online ranker evaluation”. In: *ACM International Conference on Web Search and Data Mining (WSDM)*.

Appendix A

FURTHER EXPERIMENTS

Synthetic Functions. We evaluated on a range of 16-arm synthetic settings derived from the utility-based dueling bandits setting of Ailon, Z. Karnin, and Joachims (2014). For the multi-dueling setting, we used the following preference functions:

Name	Distribution of Utilities of arms
1good	1 arm with utility 0.8, 15 arms with utility 0.2
2good	1 arm with utility 0.8, 1 arms with utility 0.7, 14 arms with utility 0.2
6good	1 arm with utility 0.8, 5 arms with utility 0.7, 10 arms with utility 0.2
arith	1 arm with utility 0.8, 15 arms forming an arithmetic sequence between 0.7 and 0.2
geom	1 arm with utility 0.8, 15 arms forming a geometric sequence between 0.7 and 0.2

Table A.1: 16-arm synthetic datasets used for experiments.

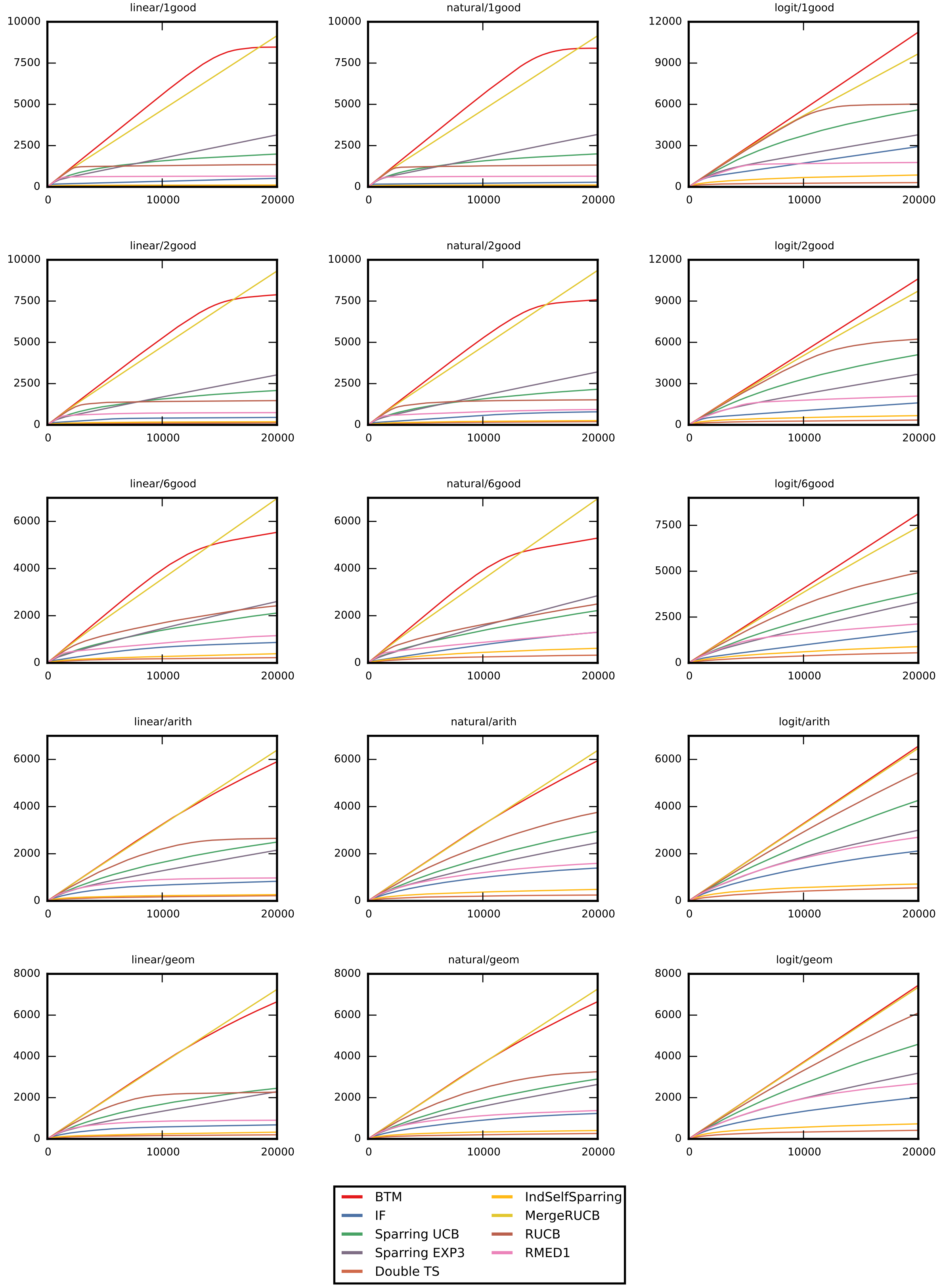


Figure A.1: Average regret vs iterations for each of 8 algorithms and 15 scenarios.

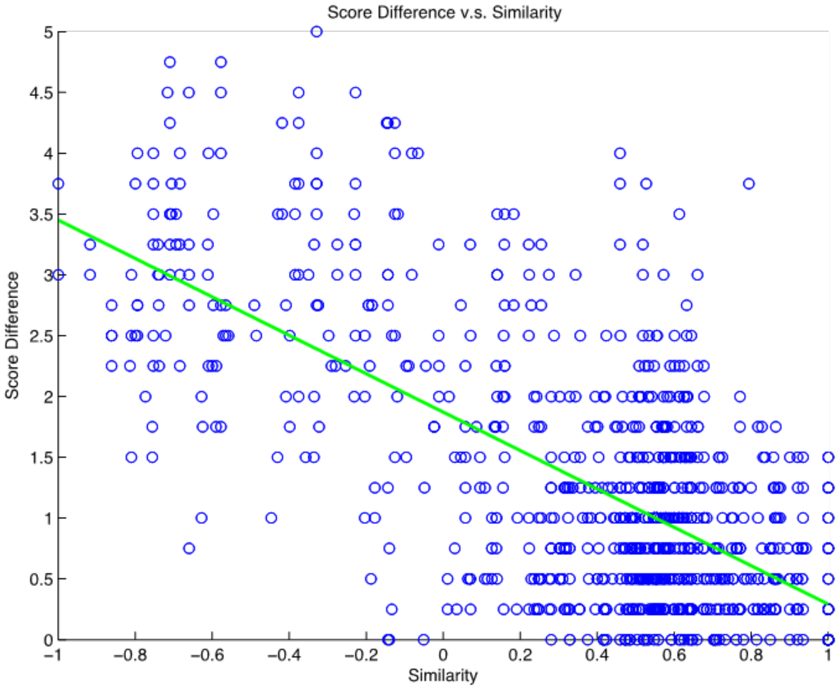


Figure A.2: Scores v.s. Similarity

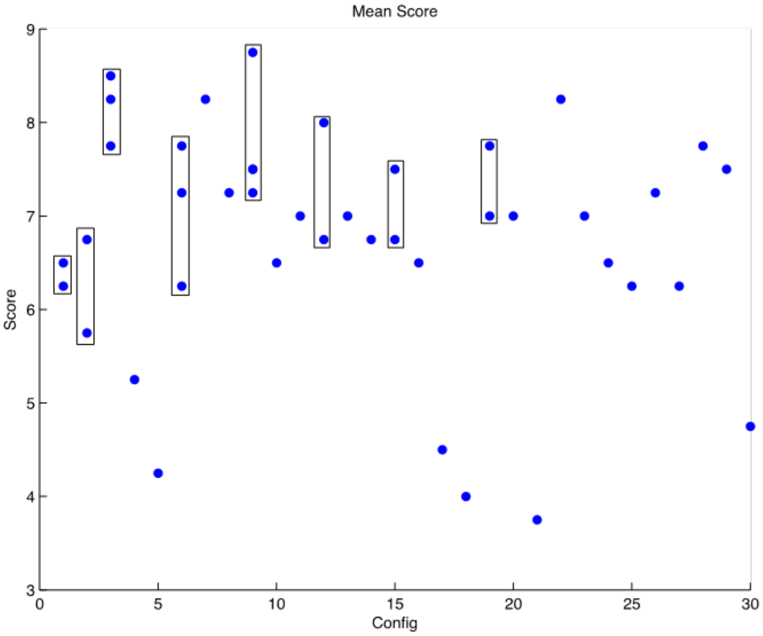


Figure A.3: Scores v.s. Configurations

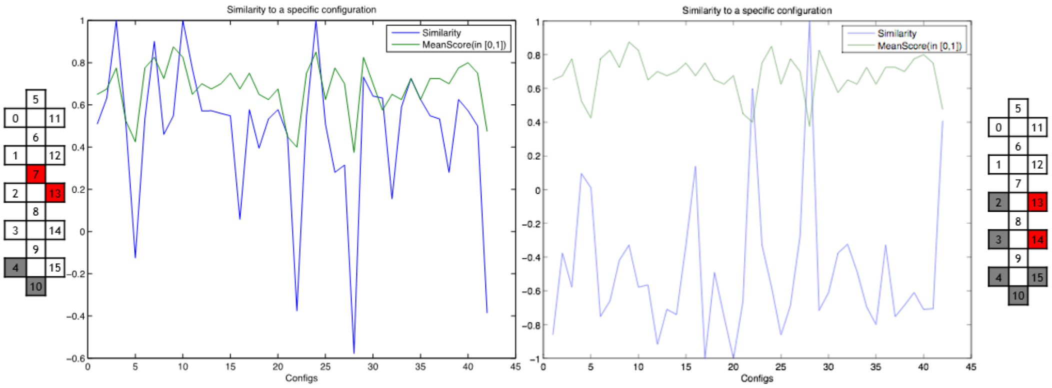


Figure A.4: Correlations between similarity measures and performance scores

Appendix B

PROOFS

B.1 Proofs for Ch3

Lemma 3. If approximate linearity holds, competing with a drifting but converging distribution of arms guarantees convergence for Thompson Sampling.

Proof. Let D_t be the drifting but converging distribution and $D_t \to D$ as $t \to \infty$. Let b_T be the drifting mean bandit of D_T after T iterations. Since D_t is convergent, $\exists T > K$ so that

$$\phi(\sup_{t>T} b_T, \inf_{t>T} b_T) < \phi(b_1, b_2)$$

where $\phi(b_1, b_2)$ is the preference between the best two arms. The mean value of feedback by playing arm i is $\phi(b_i, b_T)$. If b_T is fixed, by Lemma 1, Thompson sampling converges to the arm: $i^* = \mathrm{argmax}_i\, \phi(b_i, b_T)$. For drifting b_T, define $b^+ = \sup_{t>T} b_T$ and $b^- = \inf_{t>T} b_T$.

Thompson sampling convergence to the optimal arm implies that:

$$\phi(b_1, b^+) > \phi(b_i, b^-)$$

for all $i \neq 1$. Consider:

$$\begin{aligned} &\phi(b_1, b^+) - \phi(b_2, b^-) \\ &= \phi(b_1, b^+) - \phi(b_2, b^-) + \phi(b_1, b^-) - \phi(b_1, b^-) \\ &= \phi(b_1, b^-) - \phi(b_2, b^-) + \phi(b_2, b^+) - \phi(b_1, b^-) \\ &\geq \gamma \cdot [\phi(b_1, b_2) - \phi(b^+, b^-)] > 0 \end{aligned}$$

by approximate linearity.

So we have $\phi(b_1, b^+) > \phi(b_2, b^-)$. Since $\phi(b_2, b^-) > \phi(b_i, b^-)$ for $i > 2$. Then we have

$$\phi(b_1, b^+) > \phi(b_i, b^-)$$

holds for all $i \neq 1$. So Thompson sampling converge to the optimal arm.

Lemma 4. For the K-armed stochastic MAB problem, Thompson Sampling has expected regret: $\mathbb{E}[R_T^{\mathrm{MAB}}] = O\left(\frac{K}{\Delta} \ln T\right)$, where Δ is the difference between expected rewards of the best two arms. *Proof.* This lemma is a direct result from Theorem 2 of Agrawal and Goyal (2012) and Theorem 1 of Kaufmann, Korda, and Rémi Munos (2012).

Lemma 5. Running IndSelfSparring with infinite time horizon will sample each arm infinitely often. *Proof.* Proof by contradiction.
Let $B(x;\alpha,\beta) = \int_0^x t^{\alpha-1}(1-t)^{\beta-1}dt$. Then the CDF of Beta distribution with parameters (α,β) is

$$F(x;\alpha,\beta) = \frac{B(x;\alpha,\beta)}{B(1;\alpha,\beta)}.$$

Suppose arm b can only be sampled in finite number of iterations. Then there exists finite upper bound T_b for $\alpha_b + \beta_b$. For any given $x \in (0,1)$, the probability of sampling values of arm b θ_b greater than x is

$$P(\theta_b > x) = 1 - F(x;\alpha_b,\beta_b)$$

$$\geq 1 - F(x;1,T_b-1) = (1-x)^{T_b-1} > 0$$

Then by running IndSelfSparring, the probability of choosing arm b after it has been chosen T_b times:

$$P(\theta_b \geq max_i\{\theta_{b_i}\}) \geq \prod_i P(\theta_b \geq \theta_{b_i})$$

is strictly non-zero. That violates any fixed upper bound T_b.

Theorem 1. Under Approximate Linearity, IndSelfSparring converges to the optimal arm b_1 as running time $t \longrightarrow \infty$: $\lim_{t\to\infty} \mathbb{P}(b_t = b_1) = 1$.

Proof. IndSelfSparring keeps one Beta distribution $Beta(\alpha_i(t),\beta_i(t))$ for each arm b_i at time step t. Let $\hat{\mu}_i(t) = \frac{\alpha_i(t)}{\alpha_i(t)+\beta_i(t)}$, $\hat{\sigma}_i^2(t) = \frac{\alpha_i(t)\beta_i(t)}{(\alpha_i(t)+\beta_i(t))^2(\alpha_i(t)+\beta_i(t)+1)}$ be the empirical mean and variance for arm b_i.
Obviously, $\hat{\sigma}_i^2(t) \longrightarrow 0$ as $(\alpha_i(t)+\beta_i(t)) = (S_i(t)+F_i(t)) \longrightarrow \infty$. By Lemma 5 we have $(S_i(t)+F_i(t)) \longrightarrow \infty$ as $t \longrightarrow \infty$. That shows every Beta distribution is concentrating to a Dirac function at $\hat{\mu}_i(t)$ when $t \longrightarrow \infty$. Define $\hat{\mu}(t) = [\hat{\mu}_1(t),\cdots,\hat{\mu}_K(t)]^T \in [0,1]^K$to be the vector of means of all arms. Then $\mu = \{\mu_i = P(b_i > b_1)\}_{i=1,\cdots,K}$ is a stable point for IndSelfSparring in the K dimensional mean space.

Suppose there exists another stable point $\nu \in [0,1]^K(\nu \neq \mu)$ for IndSelfSparring, consider the following two possibilities: (1) $\nu_1 = max_i\{\nu_i\}$ and (2) $\nu_1 < max_i\{\nu_i\} = \nu_j$.

Since the Beta distributions for each arm b_i is concentrating to Dirac functions at ν_i, $P(\theta_i > \theta_j) \in [\mathbb{I}(\nu_i > \nu_j)-\delta, \mathbb{I}(\nu_i > \nu_j)+\delta]$ for any fixed $\delta > 0$ with high probability.

If (1) holds, then ν_1 will converge to $\frac{1}{2} = \mu_1$ and ν_i will converge to $P(b_i \succ b_1) = \mu_i$. Thus $\nu = \mu$. Contradict to $\nu \neq \mu$.

If (2) holds, then ν_j will converge to $\frac{1}{2} = \mu_1$ and $\nu_1 \in [P(b_1 \succ b_j) - \delta, P(b_1 \succ b_j) + \delta]$ for any fixed $\delta > 0$ with high probability. Since $P(b_1 \succ b_j) \geq \frac{1}{2} + \Delta$, $\nu_1 \in [P(b_1 \succ b_j) - \delta, P(b_1 \succ b_j) + \delta] \geq \frac{1}{2} + \Delta - \delta$. Since δ can be arbitrarily small, we have $\nu_1 \geq \frac{1}{2} + \Delta - \delta > \frac{1}{2} + \delta > \nu_j$. That contradict to $\nu_1 < \nu_j$.

In summary, $\mu = \{\mu_i = P(b_i \succ b_1)\}_{i=1,\cdots,K}$ is the only stable point in the mean space. As $\hat{\mu}(t) \rightarrow \mu$, $\mathbb{P}(b_t = b_1) \rightarrow 1$.

Define $\mathbb{P}_t = [P_1(t), P_2(t), ..., P_K(t)]$ as the probabilities of picking each arm at time t. Let $\mathbb{P} = \{\mathbb{P}_t\}_{t=1,2,...}$ be the sequence of probabilities w.r.t. time. Assume IndSelfSparring is non-convergent. It is equivalent to say that $\mathbb{P}$ is not converging to a fixed distribution. Then $\exists \delta > 0$ and arm i s.t. the sequence of probabilities $\{P_i(t)\}_t$ satisfies:

$$\limsup_{t \rightarrow \infty} P_i(t) - \liminf_{t \rightarrow \infty} P_i(t) > \delta$$

w.h.p. which is equivalent of having:

$$\limsup_{t \rightarrow \infty} \hat{\mu}_i(t) - \liminf_{t \rightarrow \infty} \hat{\mu}_i(t) > \epsilon$$

w.h.p. for some fixed $\epsilon > 0$. This violates the stability of IndSelfSparring in the K dimensional mean space as shown above. So as $t \rightarrow \infty$, $\hat{\mu}(t) \rightarrow \mu$, $\mathbb{P}(b_t = b_1) \rightarrow 1$.

Lemma 6. Under Approximate Linearity, selecting only one arm via Thompson sampling against a fixed distribution over the remaining arms leads to optimal regret w.r.t. choosing that arm. *Proof.* We first prove the results for $m = 2$. Results for any $m > 2$ can be proved in a similar way.

Consider Player 1 drawing arms from a fixed distribution L. Player 2's drawing strategy is an MAB algorithm $\mathcal{A}$.

Let $R_A(T)$ be the regret of algorithm $\mathcal{A}$ within horizon T. $B(T) = \sup \mathbb{E}[R_A(T)]$ is the supremum of the expected regret of $\mathcal{A}$.

The reward of Player 2 at iteration t is $\phi(b_{2t}, b_{1t})$. Reward of keep playing the optimal arm is $\phi(b_1, b_{1t})$. So the total regret after T rounds is

$$R_A(T) = \sum_{t=1}^{T} [\phi(b_1, b_{1t}) - \phi(b_{2t}, b_{1t})]$$

Since Approximate Linearity yields

$$\phi(b_1, b_{1t}) - \phi(b_{2t}, b_{1t}) \geq \gamma \cdot \phi(b_1, b_{2t})$$

We have

$$\mathbb{E}[R_A(T)] = \mathbb{E}\mathbb{E}_{b_{1t}\sim L}\left[\sum_{t=1}^{T}[\phi(b_1, b_{1t}) - \phi(b_{2t}, b_{1t})\right]$$
$$\geq \mathbb{E}\mathbb{E}_{b_{1t}\sim L}\left[\sum_{t=1}^{T}\gamma \cdot \phi(b_1, b_{2t})\right]$$
$$= \gamma \cdot \mathbb{E}\left[\sum_{t=1}^{T}\phi(b_1, b_{2t})\right] = \gamma \cdot \mathbb{E}[R(T)]$$

So the total regret of Player 2 is bounded by

$$\mathbb{E}[R(T)] \leq \frac{1}{\gamma}\mathbb{E}[R_A(T)] \leq \frac{1}{\gamma}\sup \mathbb{E}[R_A(T)] = \frac{1}{\gamma}B(T)$$

Theorem 2. Under Approximate Linearity, IndSelfSparring converges to the optimal arm with asymptotically optimal no-regret rate of $\tilde{O}(K\ln(T)/\Delta)$. Where Δ is the difference between the rewards of the best two arms.

Proof. Theorem 1 provides the convergence guarantee of IndSelfSparring. Corollary **??** shows one-side convergence for playing against a converging distribution.

Since IndSelfSparring converges to the optimal arm b_1 as running time $t \longrightarrow \infty$: $\lim_{t\to\infty}\mathbb{P}(b_t = b_1) = 1$. For $\forall\delta > 0$, there exists $C(\delta) > 0$ such that for any $t > C(\delta)$, the following condition holds w.h.p.: $P(b_t = b_1) \geq 1 - \delta$.

For the triple of bandits $b_1 \succ b_i \succ b_K$, Approximate Linearity guarantees:

$$\phi(b_i, b_K) < \phi(b_1, b_K) \leq \omega$$

holds for some fixed $\omega > 0$ and $\forall i \in \{2, \cdots, K-1\}$. With small δ, the competing environment of any Player p is bounded. If $\delta < \frac{\Delta}{\Delta+\omega}$, $(1-\delta)\cdot(-\Delta) + \delta\cdot\phi(b_2, b_K) < 0 = 1\cdot\phi(b_1, b_1)$. The competing environment can be considered as unbiased and the theoretical guarantees for Thompson sampling for stochastic multi-armed bandit is valid (up to a constant factor).

Then IndSelfSparring has an no-regret guarantee that asymptotically matches the optimal rate of $\tilde{O}(K\ln(T)/\Delta)$ up to constant factors, which proves Theorem 2.

B.2 Proofs for Ch5

Note All following lemmas hold for any $\varnothing \subsetneq S_0 \subseteq D$, $h \in \mathbb{R}$, $\delta \in (0,1)$, and $\epsilon > 0$.

Define

$$\hat{x}_t := \operatorname*{argmax}_{x \in S_t} \ell_t(x) \left(= \operatorname*{argmax}_{x \in M_t} \ell_t(x) \right)$$

Lemma 8 *The following hold for any $t \geq 1$:*

1. $\forall x \in D, u_{t+1}(x) \leq u_t(x)$,
2. $\forall x \in D, \ell_{t+1}(x) \geq \ell_t(x)$,
3. $\forall x \in D, w_{t+1}(x) \leq w_t(x)$,
4. $S_{t+1} \supseteq S_t \supseteq S_0$,
5. $S \subseteq R \Rightarrow R_\epsilon(S) \subseteq R_\epsilon(R)$,
6. $S \subseteq R \Rightarrow \bar{R}_\epsilon(S) \subseteq \bar{R}_\epsilon(R)$.

Proof. (i), (ii), and (iii) follow directly from their definitions and the definition of $C_t(x)$.

4. Proof by induction. For the base case, let $x \in S_0$. Then,

   $$\ell_1(x) - Ld(x,x) = \ell_1(x) \geq \ell_0(x) \geq h,$$

   where the last inequality follows from the initialization in line 11 of Algorithm 11. But then, from the above equation and line 12 of Algorithm 11, it follows that $x \in S_1$.

   For the induction step, assume that for some $t \geq 2$, $S_{t-1} \subseteq S_t$ and let $x \in S_t$. By line 12 of Algorithm 11, this means that $\exists z \in S_{t-1}, \ell_t(z) - Ld(z,x) \geq h$. But, since $S_{t-1} \subseteq S_t$, it means that $z \in S_t$. Furthermore, by part (ii), $\ell_{t+1}(\times z) \geq \ell_t(z)$. Therefore, we conclude that $\ell_{t+1}(z) - Ld(z,x) \geq h$, which implies that $x \in S_{t+1}$.

5. Let $x \in R_\epsilon(S)$. Then, by definition, $\exists z \in S, f(z) - Ld(z,x) \geq h$. But, since $S \subseteq R$, it means that $z \in R$, and, therefore, $f(z) - Ld(z,x) \geq h$ also implies that $x \in R_\epsilon(R)$.

6. This follows directly by repeatedly applying the result of part (v).

Lemma 9 *Assume that* $\|f\|_k^2 \leq B$ *and* $n_t \leq \sigma$, $\forall t \geq 1$. *If* $\beta_t = 2B + 300\gamma_t \log^3(t/\delta)$, *then the following holds with probability at least* $1 - \delta$*:*

$$\forall t \geq 1 \, \forall x \in D, \; |f(x) - \mu_{t-1}(x)| \leq \beta_t^{1/2} \sigma_{t-1}(x).$$

Proof. See Theorem 6 by Srinivas et al. (2010a).

Corollary 1 *For* β_t *as above, the following holds with probability at least* $1 - \delta$*:*

$$\forall t \geq 1 \, \forall x \in D, \; f(x) \in C_t(x).$$

Note Where needed in the following lemmas, we implicitly assume that the assumptions of Lemma 9 hold, and that β_t is defined as above.

Lemma 10 *For any* $t_1 \geq t_0 \geq 1$*, if* $S_{t_1} = S_{t_0}$*, then, for any* t*, such that* $t_0 \leq t < t_1$*, it holds that*

$$G_{t+1} \cup M_{t+1} \subseteq G_t \cup M_t.$$

Proof. Given the assumption that S_t does not change, both $G_{t+1} \subseteq G_t$ and $M_{t+1} \subseteq M_t$ follow directly from the definitions of G_t and M_t. In particular, for G_t, note that for any $x \in S_t$, $g_t(x)$ is decreasing in t, since $u_t(x)$ is decreasing in t. For M_t, note that $\max_{x' \in S_t} \ell_t(x')$ is increasing in t, while $u_t(x)$ is decreasing in t (see Lemma 8 (i), (ii)).

Lemma 11 *For any* $t_1 \geq t_0 \geq 1$*, if* $S_{t_1} = S_{t_0}$ *and* $C_1 := 8/\log(1 + \sigma^{-2})$*, then, for any* t*, such that* $t_0 \leq t \leq t_1$*, it holds that*

$$w_t(x_t) \leq \sqrt{\frac{C_1 \beta_t \gamma_t}{t - t_0}}.$$

Proof. Given Lemma 10, the definition of $x_t := \text{argmax}_{x \in G_t \cup M_t}(w_t(x))$, and the fact that, by definition, $w_t(x_t) \leq 2\beta_t^{1/2}\sigma_{t-1}(x_t)$, the proof is completely analogous to that of Lemma 5.3 by Srinivas et al. (2010a).

Corollary 2 *For any* $t \geq 1$*, if* C_1 *is defined as above,* T_t *is the smallest positive integer satisfying* $\dfrac{T_t}{\beta_{t+T_t}\gamma_{t+T_t}} \geq \dfrac{C_1}{\epsilon^2}$*, and* $S_{t+T_t} = S_t$*, then, for any* $x \in G_{t+T_t} \cup M_{t+T_t}$*, it holds that*

$$w_{t+T_t}(x) \leq \epsilon.$$

Note Where needed in the following lemmas, we assume that C_1 and T_t are defined as above.

Lemma 12 *For any $t \geq 1$, if $\bar{R}_\epsilon(S_0) \setminus S_t \neq \varnothing$, then $R_\epsilon(S_t) \setminus S_t \neq \varnothing$.*

Proof. Assume, to the contrary, that $R_\epsilon(S_t) \setminus S_t = \varnothing$. By definition, $R_\epsilon(S_t) \supseteq S_t$, therefore $R_\epsilon(S_t) = S_t$. Iteratively applying R_ϵ to both sides, we get in the limit $\bar{R}_\epsilon(S_t) = S_t$. But then, by Lemma 8 (iv) and (vi), we get

$$\bar{R}_\epsilon(S_0) \subseteq \bar{R}_\epsilon(S_t) = S_t, \tag{B.1}$$

which contradicts the lemma's assumption that $\bar{R}_\epsilon(S_0) \setminus S_t \neq \varnothing$.

Lemma 13 *For any $t \geq 1$, if $\bar{R}_\epsilon(S_0) \setminus S_t \neq \varnothing$, then the following holds with probability at least $1 - \delta$:*

$$S_{t+T_t} \supsetneq S_t.$$

Proof. By Lemma 12, we get that, $R_\epsilon(S_t) \setminus S_t \neq \varnothing$, For equivalently, by definition,

$$\exists x \in R_\epsilon(S_t) \setminus S_t \, \exists z \in S_t, f(z) - \epsilon - Ld(z, x) \geq h. \tag{B.2}$$

Now, assume, to the contrary, that $S_{t+T_t} = S_t$ (see Lemma 8 (iv)), which implies that $x \in D \setminus S_{t+T_t}$ and $z \in S_{t+T_t}$. Then, we have

$$\begin{aligned} u_{t+T_t}(z) - Ld(z, x) &\geq f(z) - Ld(z, x) && \text{by Lemma 9} \\ &\geq f(z) - \epsilon - Ld(z, x) \\ &\geq h. && \text{by (B.2)} \end{aligned}$$

Therefore, by definition, $g_{t+T_t}(z) > 0$, which implies $z \in G_{t+T_t}$.

Finally, since $S_{t+T_t} = S_t$ and $z \in G_{t+T_t}$, we can use Corollary 2 as follows:

$$\begin{aligned} \ell_{t+T_t}(z) - Ld(z, x) &\geq \ell_{t+T_t} - f(z) + \epsilon + h && \text{by (B.2)} \\ &\geq -w_{t+T_t}(z) + \epsilon + h && \text{by Lemma 9} \\ &\geq h. && \text{by Corollary 2} \end{aligned}$$

This means that by line 12 of Algorithm 11 we get $x \in S_{t+T_t}$, which is a contradiction.

Lemma 14 *For any $t \geq 1$, if $S_{t+T_t} = S_t$, then the following holds with probability at least $1 - \delta$:*

$$f(\hat{x}_{t+T_t}) \geq \max_{x \in \bar{R}_\epsilon(S_0)} f(x) - \epsilon.$$

Proof. Let $x^* := \operatorname{argmax}_{x \in S_{t+T_t}} f(x)$. Note that $x^* \in M_{t+T_t}$, since

$$\begin{aligned} u_{t+T_t}(x^*) &\geq f(x^*) && \text{by Lemma 9} \\ &\geq f(\hat{x}) && \text{by definition of } x^* \\ &\geq \ell_{t+T_t}(\hat{x}) && \text{by Lemma 9} \\ &\geq \max_{x \in S_{t+T_t}} \ell_{t+T_t}(x). && \text{by definition of } \hat{x} \end{aligned}$$

We will first show that $f(\hat{x}_{t+T_t}) \geq f(x^*) - \epsilon$. Assume, to the contrary, that

$$f(\hat{x}_{t+T_t}) < f(x^*) - \epsilon. \tag{B.3}$$

Then, we have

$$\begin{aligned} \ell_{t+T_t}(x^*) &\leq \ell_{t+T_t}(\hat{x}) && \text{by definition of } \hat{x} \\ &\leq f(\hat{x}) && \text{by Lemma 9} \\ &< f(x^*) - \epsilon && \text{by (B.3)} \\ &\leq u_{t+T_t}(x^*) - \epsilon && \text{by Lemma 9} \\ &\leq \ell_{t+T_t}(x^*), && \text{by Corollary 2 and } x^* \in M_{t+T_t} \end{aligned}$$

which is a contradiction.

Finally, since $S_{t+T_t} = S_t$, Lemma 13 implies that $\bar{R}_\epsilon(S_0) \subseteq S_t = S_{t+T_t}$. Therefore,

$$\begin{aligned} \max_{x \in \bar{R}_\epsilon(S_0)} f(x) - \epsilon &\leq \max_{x \in S_{t+T_t}} f(x) - \epsilon && \bar{R}_\epsilon(S_0) \subseteq S_{t+T_t} \\ &= f(x^*) - \epsilon && \text{by definition of } x^* \\ &\leq f(\hat{x}_{t+T_t}). && \text{proven above} \end{aligned}$$

Corollary 3 *For any $t \geq 1$, if $S_{t+T_t} = S_t$, then the following holds with probability at least $1 - \delta$:*

$$\forall t' \geq 0, f(\hat{x}_{t+T_t+t'}) \geq \max_{x \in \bar{R}_\epsilon(S_0)} f(x) - \epsilon.$$

Proof. This is a direct consequence of the proof of the preceding lemma, combined with the facts that both $S_{t+T_t+t'}$ and $\ell_{t+T_t+t'}(\hat{x}_{t+T_t+t'})$ are increasing in t' (by Lemma 8 (iv) and (ii) respectively), which imply that $\max_{x \in S_{t+T_t+t'}} \ell_{t+T_t+t'}(x)$ can only increase in t'.

Lemma 15 *For any $t \geq 0$, the following holds with probability at least $1 - \delta$:*

$$S_t \subseteq \bar{R}_0(S_0).$$

Proof. Proof by induction. For the base case, $t = 0$, we have by definition that $S_0 \subseteq \bar{R}_0(S_0)$.

For the induction step, assume that for some $t \geq 1$, $S_{t-1} \subseteq \bar{R}_0(S_0)$. Let $x \in S_t$, which, by definition, means $\exists z \in S_{t-1}$, such that

$$\begin{aligned} & \ell_t(z) - Ld(z, x) \geq h \\ \Rightarrow \ & f(z) - Ld(z, x) \geq h. && \text{by Lemma 9} \end{aligned}$$

Then, by definition of $\bar{R}_0$ and the fact that $z \in \bar{R}_0(S_0)$, it follows that $x \in \bar{R}_0(S_0)$.

Lemma 16 *Let t^* be the smallest integer, such that $t^* \geq |\bar{R}_0(S_0)| T_{t^*}$. Then, there exists $t_0 \leq t^*$, such that $S_{t_0 + T_{t_0}} = S_{t_0}$.*

Proof. Assume, to the contrary, that for any $t \leq t^*$, $S_t \subsetneq S_{t+T_t}$. (By Lemma 8 (iv), we know that $S_t \subseteq S_{t+T_t}$.) Since T_t is increasing in t, we have

$$S_0 \subsetneq S_{T_0} \subseteq S_{T_{t^*}} \subsetneq S_{T_{t^*}+T_{T_{t^*}}} \subseteq S_{2T_{t^*}} \subsetneq \cdots,$$

which implies that, for any $0 \leq k \leq |\bar{R}_0(S_0)|$, it holds that $|S_{kT_{t^*}}| > k$. In particular, for $k^* := |\bar{R}_0(S_0)|$, we get

$$|S_{k^*T}| > |\bar{R}_0(S_0)|$$

which contradicts $S_{k^*T} \subseteq \bar{R}_0(S_0)$ by Lemma 15.

Corollary 4 *Let t^* be the smallest integer, such that $\dfrac{t^*}{\beta_{t^*}\gamma_{t^*}} \geq \dfrac{C_1|\bar{R}_0(S_0)|}{\epsilon^2}$. Then, there exists $t_0 \leq t^*$, such that $S_{t_0+T_{t_0}} = S_{t_0}$.*

Proof. This is a direct consequence of combining Lemma 16 and Corollary 2.

Lemma 17 *If f is L-Lipschitz continuous, then, for any $t \geq 0$, the following holds with probability at least $1 - \delta$:*

$$\forall x \in S_t, f(x) \geq h.$$

Proof. We will prove this by induction. For the base case $t = 0$, by definition, for any $x \in S_0$, $f(x) \geq h$.

For the induction step, assume that for some $t \geq 1$, for any $x \in S_{t-1}$, $f(x) \geq h$. Then, for any $x \in S_t$, by definition, $\exists z \in S_{t-1}$,

$$\begin{aligned} h &\leq \ell_t(z) - Ld(z, x) \\ &\leq f(z) - Ld(z, x) && \text{by Lemma 9} \\ &\leq f(x). && \text{by } L\text{-Lipschitz-continuity} \end{aligned}$$

Proof.[Proof of Theorem 5] The first part of the theorem is a direct consequence of Lemma 17. The second part follows from combining Corollary 3 and Corollary 4.