# Sound Relaxation: Soundscape Exploration using Reinforcement Learning

by

Yourui Guo

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

University of Alberta

# Abstract

High stress levels and depression are commonly observed in hospitalized patients, which may negatively impact them on recovery after surgery. Sound therapy has been widely used for its effectiveness in increasing relaxation and reducing stress levels, and one could also fine-tune music features such as pitch, volume and reverb to improve sleep quality and reduce anxiety. Our sound therapy project aims to select and fine-tune a soundscape that maximizes a subject's relaxation level. We simplified this problem and formulated it as a reinforcement learning problem where the goal is to select a soundscape, and determine the best volume setting for each of its component sounds. For the virtual subject experiment, we tested RTA*, UCT and few baseline algorithms. We also conducted experiments on human subjects, divided into a treatment group that listen to sounds selected by the UCT algorithm and a control group that listen to sounds selected by a human experimenter. The UCT algorithm displayed the same performance as the human, suggesting that sound therapy can be automated using UCT.

# Preface

This thesis is an original work by Yourui Guo. The research project, of which this thesis is a part, received research ethics approval from the University of Alberta Research Ethics Board, Project Name "Sound Relaxation: Soundscape Exploration using Reinforcement Learning", No. Pro00097850, June 17, 2020.

*Act without action. Pursue without interfering. Taste the tasteless.*

– Tao Te Ching

为无为，事无事，味无味。

– 道德经

# Acknowledgements

I would like to express my utmost gratitude to my supervisors Dr. Abram Hindle, Dr. Nathan Sturtevant and Dr. Michael Frishkopf for their constant support, guidance and motivation. It would never have been possible for me to take this work to completion without their incredible support and encouragement. Dr. Hindle guided me throughout every step of the work and gave helpful suggestions to my research and writing skills. Dr. Sturtevant persistently provided his valuable and constructive advice during the development of this research work. Dr. Frishkopf supported this project by offering insights from a musical perspective, and provided me with encouragement and patience throughout the duration of this project. I extend my sincere thanks to Dr. Martha Steenstrup for her invaluable advice and fruitful discussions.

Special thanks to KIAS and the University of Alberta for their generous financial support and Deep Learning for Sound Recognition (DLSR) group for bringing much insight into the problems in this work.

I would like to thank the extremely bright individuals who work in the Software Engineering Research Lab.

This work could not have been completed without the support and great love of my family, my parents, and my husband.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

In order to reduce the subjects' stress and anxiety level, a learning system can be built to produce sounds that are relaxing to the subjects via their implicit feedback. High stress levels, anxiety, insomnia and depression are commonly observed in critically ill patients that may negatively impact their recovery from surgery and illness [30, 10, 37]. There is mixed evidence [29, 35, 39, 1, 13] showing that sound therapy is effective at increasing listeners' relaxation levels. It is also an inexpensive, low-risk, and a highly accepted tool for addressing mental health issues caused by stress [39]. In addition, research [29] suggests that personalized sound therapy sessions are the most effective form of sound therapy. The effectiveness of sound therapy can be affected by many factors such as medical condition, age, ethnic background, listening history, musical preferences [33], previous music experience [38], and the type of stress [1, 13, 4]. Different measurements of sound therapy experiments might lead to opposite conclusions [39]. Some experiments suggest that various types of sound might have different effects on subjects' relaxation level [35, 17, 33]. For example, a group that listened to the sound of bubbling water had lower concentrations of cortisol (a stress hormone) than a group that listened to relaxing music [35]. Another study illustrated that the genre of music is a significant factor for stress reduction in contrast to music preference [17]. Some evidence indicates that music selected based on personal preference has a great effect on decreasing anxiety level [33].

If we apply sound therapy to critically ill patients, the challenge arises that

these patients might not be able to give verbal feedback via self-reporting. The question then arises whether it is possible to design a system capable of producing an effective soundscape to reduce stress, in such a manner as to require minimal or no conscious feedback from the patient.

Our goal is to build a learning system that is able to automatically select and fine-tune soundscapes to increase listeners' relaxation level, with minimal patient feedback. We include various categories of natural sounds to this system as acoustic nature sounds showed a significant effectiveness in decreasing subjects' anxiety [35]. With the assumption of requiring minimal user feedback, the system should learn to determine the sound categories that the patient likes and dislikes given minimal information gained from the patient.

We build two separate systems based on two Reinforcement Learning (RL) problem models. These RL models can be applied to this task by training an agent to take actions seeking higher reward from the environment [34]. For each system, we develop several algorithms to explore soundscapes on the basis of implicit and subjective responses. One system is built on a Markov Decision Process (MDP) model [34] while the other is built using a multi-armed bandit [2] model. We propose an algorithm that is based on Real-Time Heuristic Search (RTA*) [24] to solve the MDP problem, and another algorithm using Upper Confidence Bound for trees (UCT)[2] for the multi-armed bandit problem. We also implement some baseline algorithms for comparison.

In this work we specify a sound category for each sound, which means every sound belongs to only one sound category and each sound category is comprised of multiple sounds. Every sound is at one of three volume levels: silent, low and high. The agent running the UCT algorithm chooses a vector of volume levels for each component sound in a category since it assumes the state space has a hierarchical structure. In other words, the volume level of all sounds under a selected category can be low or high while rest of sounds are silent. Unlike UCT, the agent running RTA* ignores sound categories and might explore all combinations of the volumes of sounds.

RTA* is a path-finding algorithm that assumes the goal is the most relaxing soundscape for the subject. The goal is not guaranteed to be unique, which

means there might be multiple goals or the goal might change over time. For simplicity, we assume the goal is steady and unique for RTA*. The agent running RTA* starts with any soundscape at any volume level, set randomly. It aims at determining the goal soundscape by exploring various soundscapes with different combinations of volume for each of its component sounds. But because RTA* does not organize sounds hierarchically, it cannot determine the goal soundscape within a reasonable time period. Furthermore, RTA* is not practical for real subjects when the subjects prefer multiple soundscapes or they change their preference over time. Thus, the UCT algorithm is introduced to address the above problems and balance the exploration-exploitation trade-off. We model each participant as having a potential preference score for each sound category and volume level; the purpose of the UCT algorithm is determining the score of all sounds and volume levels. UCT keeps exploring when the soundscape has an inaccurate preference score or a small number of visits. The drawback of UCT is that when the soundscape with potential maximum score is found, it keeps sampling other soundscapes for exploration, which might annoy the subjects.

We conducted experiments on both virtual subjects and real subjects. The result of the experimental testing on virtual subjects indicates that the multi-armed bandit model performs better than the MDP model. Overall, the results show that both RTA* and UCT algorithms perform better than baseline algorithms, and UCT performs better than RTA*. Since UCT outperforms RTA* in virtual subject experiments, we measured the system running UCT algorithm on real subjects where the subjects were divided into two groups for comparison. The treatment group listened to sounds selected by the UCT algorithm and the control group listened to sounds selected by the experimenter. Self-reported responses from both groups indicate that subjects' stress levels were significantly reduced. We conclude that the sound therapy can be automated using the UCT algorithm, as it displayed the same performance as the human.

# Chapter 2

# Background and Related Work

In this chapter, background and literature related to Reinforcement Learning, music therapy, and music recommendation systems are introduced. In particular, we present some algorithms for heuristic search such as A* and RTA*, as well as bandit algorithms such as UCB1 and UCB applied to trees (UCT). In addition, we provide related work to demonstrate the effectiveness of music therapy and state-of-the-art of music recommendation systems in this chapter.

## 2.1 Reinforcement Learning

Reinforcement Learning (RL) [34] is an area of Machine Learning that trains from experience, making a sequence of decisions based on past experiences in an attempt to maximize cumulative reward from the environment. The agent-environment interface can be described as follows:

1. **Agent**: The agent takes actions and interacts with the environment. At each step, the agent receives an observation from the environment by taking an action.

2. **Environment**: The environment responds to each action with a reward and presents a new state to the agent.

3. **Reward**: Reward is a number given by the environment. The agent tries to maximize the total amount of reward it receives.

4. **State**: observations the agent receives from the environment.

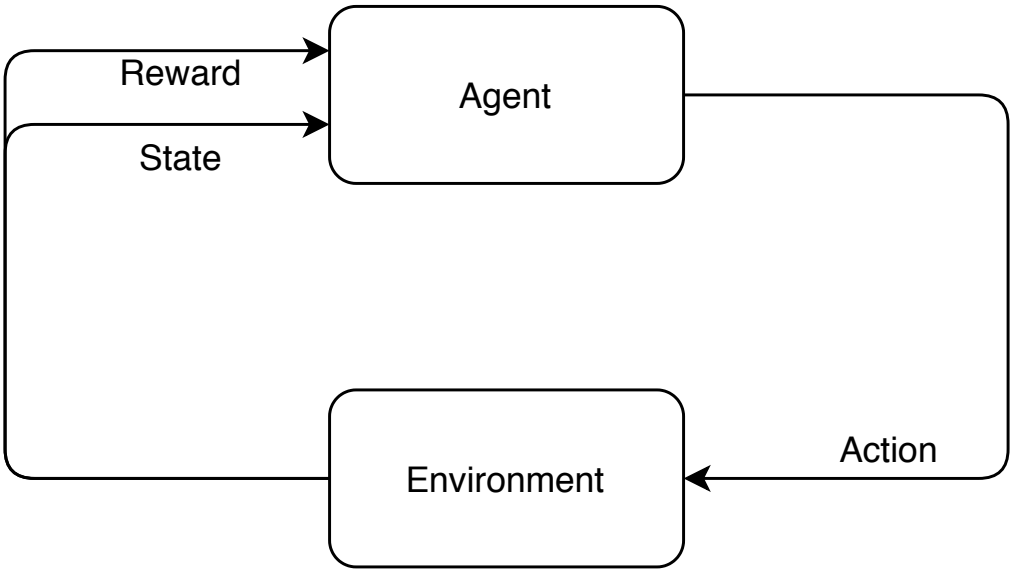



Figure 2.1: Agent-Environment Interface of Reinforcement Learning System

5. **Action**: Each state is associated with a set of possible actions that the agent can perform.

In addition, there is other terminology we will use in the following sections. A **policy** [34] maps the states of the environment to actions the agent can take. The **reward signal** [34] defines the goal in a RL problem. This is a number that indicates a better or worse event for the agent. The **value function** [34] represents the total amount of reward the agent expects to accumulate starting from a particular state: $V_s^\pi$ is expected reward from following policy $\pi$ at state s.

### 2.1.1 Heuristic Search

Heuristic search takes advantage of heuristic information where it approximates the cost of reaching the goal from a state in search problems. The agent starts in one state and aims at determining a path to a goal state in the state space. A heuristic $h(s)$ is an estimate of the cost of a path from the current state $s$ to a goal state. A heuristic is admissible if it never overestimates the cost to get to the goal. $g(s)$ is the actual cost of the path from the start state to current state $s$. $f(s)$ is an evaluation function that estimate the total cost

to the goal from state $s$. It varies according to different search algorithms. For example, the evaluation function in A* algorithm is $f(s) = h(s) + g(s)$, and the evaluation function in greedy best-first search is $f(s) = h(s)$.

A state is **expanded** if its successors are generated in the state space. The **open list** is a queue that stores and sorts the states that have been generated but not expanded. If a state is expanded, it will be removed from the open list and put into the closed list. The closed list maintains the states that have been expanded.

There are many algorithms in heuristic search, such as A*[15], Iterative Deepening A* (IDA*) [23], weighted A* [12], and Real Time A* (RTA*) [24].

A* is initialized with an empty closed list and an open list that includes the start state. Every time the agent reaches a new state, adjacent states that are not on the closed list are added to the open list. The state on the open list with the minimum $f(s)$ is expanded, removed from the open list and added to the closed list. The algorithm terminates when the goal is reached (assuming there exists a goal in the state space), or the open list becomes empty.

**Algorithm 1** A*

```
1:  openlist[0] ← start
2:  while openlist != NULL do
3:      state ← min(openlist)
4:      add state to closedlist
5:      if state is goal then
6:          break
7:      while action in next actions do
8:          next ←applyAction(action)
9:          if next in closedlist then
10:             continue
11:         else if next in openlist then
12:             new_cost ← gcost(state) + Cost()
13:             if new_cost < gcost(next) then
14:                 gcost(next) ← new_cost
15:         else
16:             add next to openlist
```

A* has a limited performance when the environment cannot provide the knowledge that A* requires. Since A* finds a path offline before taking ac-

tions, it might fail to take an action within a limited time when the heuristic information is not accurate enough. In contrast to A*, Real Time A* (RTA*) is able to explore the state space by interleaving actions and thinking.

Real Time A* (RTA*), is commonly used in real-time planning and learning problems such as path planning for mobile robots. As RTA* doesn't require offline exploration to find the path to the goal before execution and allows the agent to search in real time, it is suitable for the tasks where the agent interacts in real time with an unknown environment within a limited time period. Bulitko and Lee [5] studied real time heuristic search algorithms for learning and planning problems. They developed an algorithm called Learning in real-time search (LRTS) that unifies many prior algorithms such as LRTA* [24], $\epsilon$-LRTA*, SLA* [31], and $\gamma$-Trap [6]. algorithm. Cannon and Rose et al [7] proposed an algorithm called Partitioned Learning Real-time A* (PLRTA*) adapting the LSS-LRTA* [22] algorithm for real-time heuristic searching in a high dimensional space. PLRTA* allows the agent to perform well in dynamic motion planning. Howlett and McLain et al. [16] present a path-finding algorithm based on LRTA* for unmanned air vehicle (UAV) with a specific sensor footprint. This algorithm enables the agents to quickly determine short-distance paths. Kim [20] proposed the DFS-RTA* algorithm based on depth-first search that adapts real-time path-finding methods for fast backtracking. This prior work implies the efficacy and efficiency of real time heuristic search for addressing real-time path finding problems in unknown environments.

While A* finds an entire path offline before execution, RTA* approaches the problem by making a sequence of decisions to the goal in a single trial [24]. The challenge of RTA* is to avoid infinite loops when backtracking out of local minima. The principle of backtracking to a previous state is when the estimated cost of going forward from the previous state $h(prev)$ is less than the estimate cost of going forward from the current state $h(current)$.

$$h(prev) < h(current)$$

The estimate of the previous state $h(prev)$ at the next step is composed of its

original estimate of solving the problem $h_{original}(prev)$ and the actual cost of returning from the current state $g(current)$.

$$h(prev) = g(current) + h_{original}(prev)$$

The agent is allowed to backtrack when this assumption holds: the estimate cost of previous state $h_{original}(prev)$ plus the cost from the current state to the previous state $g(current)$ is less than the estimate cost of going forward from the current state $h(current)$.

At the same time, the estimate cost of the previous state is updated with the second best $f(s)$ from its neighbors. This is because the estimate cost of solving the problem from the previous state can be represented by the estimate of the previous state's second best neighbor since its best adjacent state was chosen as the current state at the previous step. Unlike A*, the interpretation of $g(s)$ in RTA* is the actual cost relative to the current state and is independent of the initial state. Further details about how RTA* is adapted into the system are introduced in Chapter 5.2.

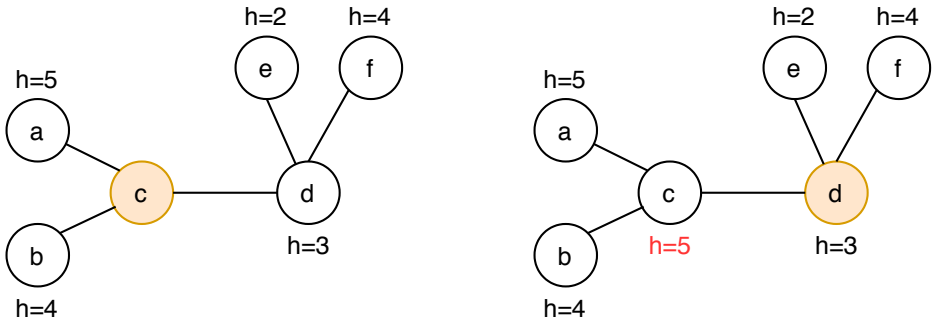



Figure 2.2: An example of RTA* where the agent fails to backtrack to a previous state

An example of RTA* is explained as follows. Figure 2.2 illustrates a situation in which an agent running RTA* could not backtrack to a previous state. We assume that the $g(s)$ of each edge is 1, and that the start state is state c. Since the adjacent state d has the minimum $h(s)$, the agent moves to state d and the $h(s)$ of c is updated with the second best $f(s)$ from state b. The

agent will not backtrack to state c because it has the highest $h(s)$ among all neighbors (state c, e and f).

### 2.1.2 Bandit Algorithms

.

Bandit algorithms are commonly applied to decision-making problems to balance the exploration-exploitation trade-off. The general bandit framework process is to choose an arm to pull based on a bandit algorithm [1], observe the reward after pulling the selected arm, and update its estimated reward.

A multi-armed bandit model can be described as a set of tuples $< A, X >$ where $A$ represents the action and $X$ represents the reward. The reward probability distribution of the k arms is $\{P_1, P_2, ..., P_k\}$ with respective means $\{\mu_1, \mu_2, ..., \mu_k\}$. The expected reward $\mathbb{E}[r|a] = \mu_a$. The optimal arm is argmax over i of $\mu_i$. The definition of the regret $R_n$ from the book [25] is as following:

> The regret of a learner following policy $\pi$ is the difference between the total expected reward using policy $\pi$ for n rounds and the total expected reward that the learner actually collected for n rounds.

The regret is a measure of loss over a policy $\pi$.

$$R_n(\pi) = n\mu^* - \mathbb{E}\left[\sum_{t=1}^{n} X_t\right]$$

where $X_t$ is the reward at time step $t$ and $X_t$ are 1-subgaussian random variables.

We define sub-optimal gap $\Delta = \mu^* - \mu_a$ as the difference between the expected reward of optimal arm and the selected arm.

$$T_a(t) = \sum_{s=1}^{t} \mathbb{I}\{A_s = a\}$$

is the number of times that action $a$ was selected by the learner after t steps. Let $\mathbb{I}\{A_s = a\}$ be a binary variable (1 or 0) to indicate whether or not action

[1] the player pulls arms in a bandit machine for payoffs

$a$ is chosen at time step $s$. Thus, the regret can be decomposed as:

$$R_n = \sum_{a \in A} \Delta_a \mathbb{E}[T_a(n)].$$

The policy UCB1 (Upper Confidence Bound) [3] is a bandit algorithm that selects the next arm to play based on the sequence of previous actions and obtained rewards. It has a good performance on multi-armed bandit problem and does not depend on knowledge of sub-optimal gaps.

**Algorithm 2** UCB1(k, $\delta$)

1: $t \leftarrow 1$
2: **while** t $\leq$ n **do**
3: *choose action* $A_t = argmax_i UCB(t-1, \delta)$
4: *Observe reward* $X_t$ *and update upper confidence bound*
5: $t \leftarrow t + 1$

UCB1 is optimistic about uncertainty and prefers to explore actions without an accurate estimate value [25]. The strategy of being optimistic works because if the optimism is justified, the selected action is the optimal action; if the optimism is not justified (the learner assumes the action should receive larger reward but in fact does not), the learner will soon learn the true payoff of the action [25]. A smaller sub-optimal gap $\Delta$ makes the algorithm harder to determine the optimal action as the algorithm requires more sampling to distinguish the optimal action from all sub-optimal actions when the expected reward of a sub-optimal action is close to that of the optimal action.

The following derivation closely follows the outlines provided in the book *" Bandit Algorithm"* by Szepesvari and Lattimore [25]. Let $\hat{\mu} = \sum_{t=1}^{n} \frac{X_t}{n}$ where $n$ is the number of selected times, and $\mathbb{P}(\hat{\mu} \geq \epsilon)$ as the probability of $\hat{\mu}$ greater than $\epsilon$ , and we can get

$$\mathbb{P}(\hat{\mu} \geq \epsilon) \leq \exp\left(-\frac{n\epsilon^2}{2}\right)$$

according to Hoeffding's inequality since $X_t$ are 1-subgaussian independent random variables. A random variable $X$ has 1-subgaussian distribution when $\mathbb{E}[X] = 0$ and $\mathbb{E}[X^2] \leq 1$. Let $\delta = \exp\left(-\frac{n\epsilon^2}{2}\right)$, and then

$$\mathbb{P}(\hat{\mu} \geq \sqrt{\frac{2}{n}\log(\frac{1}{\delta})}) \leq \delta$$

Since $T_a(t-1)$ represents the number of times that action $a$ was selected, we can estimate the highest possible mean of action $a$ as:

$$\hat{\mu}_a(t-1) + \sqrt{\frac{2}{T_a(t-1)} \log(\frac{1}{\delta})}$$

In summary, the UCB1 algorithm chooses the action $a$ at time step $t$ according to:

$$A_t = argmax_i(\hat{\mu}_a(t-1) + \sqrt{\frac{2}{T_a(t-1)} \log(\frac{1}{\delta})})$$

UCB applied to trees (UCT) [21] is a popular algorithm in Monte-Carlo Tree Search (MCTS). The purpose of UCT is determining the true value of the actions that might be taken in each state. It treats the bandit problem as a separate multi-armed bandit for each node in a tree structure. UCT applies the UCB1 algorithm to action selections in rollout-based planning [21] where the action selected in state $s$, at depth d aims at maximizing $Q_t(s,a,d) + c_{N_{s,d}(t),N_{s,a,d}(t)}$. $Q_t(s,a,d)$ is an estimated value of action $a$ at time step $t$ and depth $d$ in state $s$. $N_{s,d}(t)$ is the number of times that state $s$ has been visited before $t$ steps, and $N_{s,a,d}(t)$ is the number of times that action $a$ has been chosen in state $s$ at depth $d$ before $t$ steps. The reward of node $s$ is the discounted cumulative rewards of the path originated at the node $s$.

In UCT, it traces the best reward of $\mu_{i,T_i(t-1)}$ for each arm and selects the arm with the best UCB:

$$A_t = argmax_i(\hat{\mu}_{i,T_i(t-1)} + c_{t-1,T_i(t-1)})$$

where $c_{t,s}$ is $2C_p\sqrt{\frac{\log t}{s}}$ and $C_p$ is an appropriate constant parameter. When the reward value is in the range of $[0,1]$, $C_p = \frac{1}{\sqrt{2}}$ satisfies the Hoeffding's inequality [21]. Kocsis and Szepesvari [21] also proved that the bounds on the regret of UCB1 hold when applied to non-stationary bandit problem and the failure probability converges to zero at a polynomial rate as the number of episodes increases to infinity. Furthermore, section 5.3 introduces specific utilization of UCT in this work.

## 2.2 Music Recommendation

Several research studies center on the field of music recommendation using assessments of mood or emotion via user interactions. Some of these studies [11, 14] require prior knowledge, such as user preferences for different emotions or situations. Dhahri and Kazunori et al.[11] proposed an autonomous and adaptive song recommendation system with implicit user input feedback using a Reinforcement Learning framework with softmax selection. The system includes a personalized song map that contains users' preferred songs with metadata, and a RL framework that selects songs based on users' mood and implicit input. Griffiths and Cunningham et al [14]. sought to generate music playlists automatically using quantified human emotion data gathered from a range of sensors. They determined human emotional state by analyzing physiological and contextual data via a Fuzzy Inference System, while defining musical categories by analyzing extracted audio features. Isuru and Cohen et al. [18] provided a perspective on exploring the "sweet spot" in the soundscape using a Deep Q-Network RL agent to improve mental health.

Some research addresses the problem of playlist generation using Reinforcement Learning. DJ-MC [26] is a music recommendation system that formulates the problem as a Markov Decision Process and generates a personalized playlist within a single listening session of 25-50 songs using a model-based RL approach. In this study, the Markov state includes an ordered list of songs in the playlists, and the action is a selection of next songs to play. It also defines a deterministic transition function $P$ that represents the probability of transitioning from the current state to the next state. A song is modeled by spectral auditory descriptors where rhythmic characteristics, loudness and the change over time are included. Each song can be factored as a 34-dimensional descriptor vector. A reward function $R_s(a)$ is applied to model the human listening experience and it is composed of a binary feature vector $\theta_s(a)$ and a weight vector $\phi_s(u)$. The reward is: $R_s(a) = \phi_s(u) \cdot \theta_s(a)$, where the parameter of $\phi_s(u)$ should be learned for each new user. The reward over transition $R_t(a_i, a_j)$ represents the experience of listening to a song $a_j$ after a song $a_i$.

Similar to $R_s(a)$, $R_t(a_i, a_j)$ is composed of a binary feature factor $\theta_t(a_i, a_j)$ and a weight vector $\phi_t(u)$ that depends on users where $R_t(a_i, a_j) = \phi_t(u) \cdot \theta_t(a_i, a_j)$. To avoid oversized transitions of feature vectors, only 10-percentile bins of the same song descriptors are included in $\phi_t(u)$. The initial song preferences and initial transition preferences are generated according to a list of favourite songs provided by the user. To generate the next action based on the transition reward function, a tree-search heuristic for planning is applied. It clusters songs based on their features to reduce the search time complexity. This study is evaluated with human participants using real data and songs. Binary listener feedback is provided as "like" or "dislike". According to their results, DJ-MC outperforms the baseline algorithm significantly in terms of cumulative rewards.

Wang and Yi et al.[40] study bandit approaches for a balanced exploration-exploitation trade-off. Their results indicate that LinUCB [3] and Bayes-UCB [19] perform well in terms of the user ratings. Bayes-UCB is adopted to this study where the payoff $U_i$ is viewed as a random variable and the posterior distribution of $U_i$ given the history payoffs of $D$ is used as the upper confidence bound (UCB) in Bayes-UCB. In this study, music is represented by a feature vector x, and user preference $U_c$ can be formed as $U_c = \theta' \cdot x$. The objective is to compute the posterior distribution of parameters $U_c$ given the history data. It also defines the novelty to describe the repetition of the songs at proper frequencies. The recommendation is generated according to Zipf's law [41] where the users' listening frequencies are ranked in decreasing order. A combined model is applied to represent user feedback where the users' preferences and novelty are both considered. To enable a responsive and efficient model, this research used piecewise linear approximation to represent the irregular payoff $U_i$. In the experiments on human subject, they compared Bayes-UCB with random, greedy algorithms as well as LinUCB. Each song is a 30-second audio clip, transformed to a feature vector using the MARSYAS library [36]. The user provides feedback using a scale number to indicate the listening experience of each song. The results show a great effectiveness in music recommendation using Bayes-UCB compared to other baseline algorithms. Further study may

include extending Bayesian models to hierarchical Bayesian models as well as considering more factors besides music features such as mood, genre and diversity.

Latent Markov Embedding (LME) is used by Chen and Moore et al. [8] for playlist generation, where the problem is formulated as a regularized maximum-likelihood embedding of Markov chains in Euclidean space. LME is a machine learning algorithm that doesn't require songs descriptive features to generate playlists. The goal of this study is to estimate a generative model of playlists that helps to explore new playlists efficiently, where the playlist is modeled as a path through a latent space. Latent space refers to the multi-dimensional space that contains variables inferred from other observed variables. In this study, they assume that Euclidean distance between songs can indicate the transition probabilities and determining the location of each song is an important problem to address. A dual-point model is introduced where a pair of points $(U(s), V(s))$ is applied to represent each song $s$. $U(s)$ is the "entry point" of song $s$ that models the interface to the previous song and $V(s)$ is the "exit vector" that models the interface to the next song. Calculating the vector of $(U(s), V(s))$ becomes a maximum-likelihood problem given training samples of playlists. A norm-based regularizer is added to the log-likelihood objective to avoid over-fitting. To address the optimization problem of the LME model, stochastic gradient training and landmarks heuristics are applied. Landmarks are randomly chosen and each song is assigned to a nearby landmark. Landmarks reduce the complexity of search where a subset of songs near a landmark is added to the successors of each song. The experiment analyzes performance of LME compared to n-gram baselines. Results show that LME outperforms bigram models where the embedding of the dual-point model qualitatively reflects the intuition of musical similarity.

## 2.3 Music Therapy

Music therapy is the use of sounds to improve the subject's physical and mental health, especially to increase relaxation and decrease anxiety and stress levels.

Studies [38, 39, 30, 35] have shown significant effectiveness of music therapy for increasing relaxation. In this section, we present a few works in the area of music therapy to illustrate how different methods and experimental settings affect the subject's relaxation level, defined using a variety of measurement techniques.

In order to assess the relative efficacy of music therapies based on different formulations of personal preference, Walworth [38] investigated the difference in subjects' anxiety level after listening to music selected by three different strategies: no music, music selected by personal preference, and music genre listed as relaxing by the subject. The results showed that using patients' preferred music was more effective than other options in reducing anxiety level in a hospital setting. In addition, a specific preferred song was as effective as a preferred music genre or artist.

Wang and Kulkarni et al. [39] studied the effectiveness of music therapy in decreasing anxiety before surgery. Each patient was assigned one of two groups: Subjects in group 1 listened to patient-selected music; Subjects in group 2 received no intervention. The State-Trait Anxiety Inventory (STAI) self-reports indicated that group 1 displayed a lower anxiety level compared with the control group. In contrast, there were no differences between the two groups regarding physiological outcomes such as blood pressure (BP), heart rate (HR), electrodermal activity (EDA), or neuroendocrina variables such as cortisol, epinephrine, and norepinephrine.

Robb and Nichols et al.[30] investigated the effects of **music assisted relaxation (MAR)** interventions applied to surgical patients. Subjects ranging in age from 8 to 20 years were divided into an experimental and control groups. Subjects in the experimental group received MAR intervention that includes music listening, deep diaphragmatic breathing, progressive muscle relaxation and imagery, while the control group received only standard preoperative interventions. Results indicated that subjects who received MAR interventions experienced a significant decrease of anxiety measured by State-Trait Anxiety Inventory for Children (STAIC). No significant difference was found for either group regarding physiological measures such as heart rate, respiration rate,

blood pressure and temperature.

Thoma and Marca et al. [35] examined the effect of listening to music prior to a standard stressor among healthy participants in a laboratory setting. The hypothesis of the study was that participants who listened to relaxing music prior to a stressor such as Trier Social Stress Test (TSST) would have a different stress responses than the non-music control group. Stress responses were measured with cortisol, salivary, alpha-amylase, heart rate, respiratory sinus arrhythmia, and subjective perception of stress. Participants were divided into three groups: an experimental group that listened to relaxing music, a non-music acoustic control group that listened to natural sounds, and a non-acoustic control group. Results indicated that pre-stress music listening might not affect the physiological stress response, but might facilitate autonomic recovery from a stressor compared with the non-music and non-acoustic control groups. The recovery data is the difference between the first baseline value after the stressor and the peak values after the stressor.

# Chapter 3

# Problem Definition

As introduced in the Chapter 1, the purpose of this work is to automatically select sounds and volume levels that makes listeners feel relaxed without any prior knowledge of the subject's preferences or listening history. A learning system can be built to explore the space of sounds and discover which ones increase the patient's relaxation level, and decrease stress and anxiety. Sound features such as volume, filter, reverb and pitch might also influence stress reduction [28, 9]. The system should be able to fine-tune such features to maximize the listener's relaxation level. The input of this system is subject feedback, including autonomic bio-signals such as heart rate, blood pressure or EEG signals, as well as minimal conscious response to sounds. The output of this system is a soundscape: a mix of sounds, each at a particular volume level, that is played to the subject.

Figure 3.1 illustrates a model, where subject feedback is converted to reward. The selection of sounds can be viewed as the action taken by the agent. We assume that the algorithm acts as the agent, while the environment consists of either a real or virtual subject. With the general idea of developing the system, we expect that the algorithm makes a decision on selecting sounds and sound features to help increase reward, based on the subject's feedback. Subject feedback can be also used as the input to the agent to help it make better decisions.

In this thesis, the subject feedback is the listener's reaction to the sounds, where the listener presses a "next soundscape" button to indicate dislike of

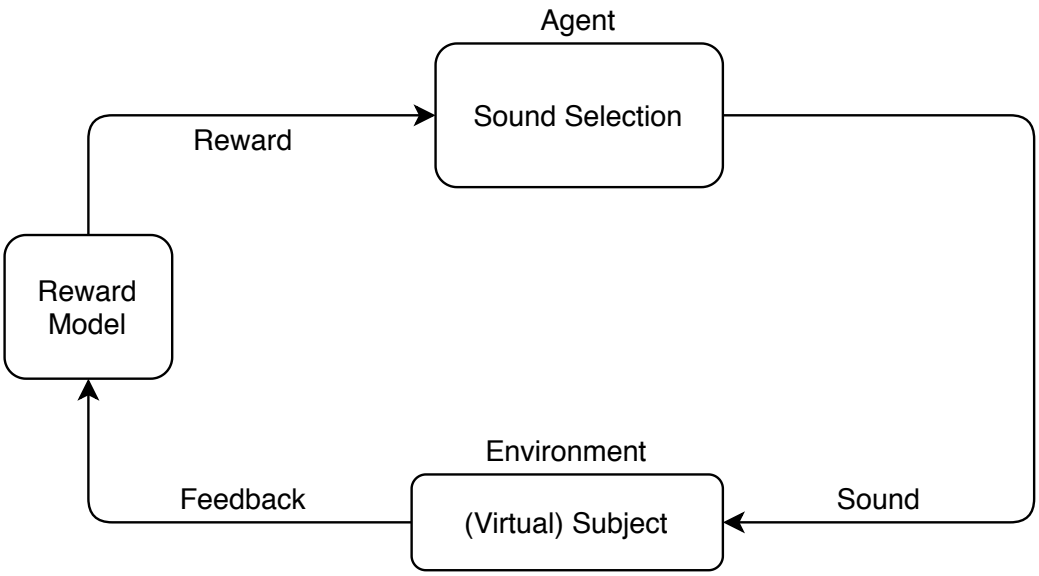



Figure 3.1: The Architecture of Soundscape Exploration System built on RL System

the current-playing sounds. We assume that people are satisfied with the current-playing sounds when they keep listening and do not press this button to switch to the next one. Thus, the longer the real subject listens, the higher the reward. In order to test the system, we also introduced a virtual subject to generate virtual reward in the experiment.

# Chapter 4

# Environment Model

A soundscape is a set of sounds that are mixed together. It includes synthesized sounds and natural sounds such as bird singing and rain. We choose a number of distinctive environmental sonic categories, such as ocean, forest, and city, and then select sounds that match those categories. In this section, we define two types of state spaces along with one virtual reward model and one real-world reward model.

## 4.1 State Space

For simplicity we assume the state space is composed of volume levels for all sounds. The state space is multi-dimensional where each dimension corresponds to one sound and the value in that dimension corresponds to a volume level of the sound. The volume of each sound can be modified independently regardless of its category. Each state represents a vector of volume values and each volume value corresponds to one of the sounds comprising the soundscapes.

### 4.1.1 Grid-world Model

We define the following tuple to represent the Grid-World model for this problem: $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{D}, (\mathcal{R}_d)_{d \in \mathcal{D}} \rangle$ where $\mathcal{S}$ is a $k$-dimensional state space and $k$ indicates the number of sounds in the entire soundscape; $\mathcal{A}$ is an action space where each action is the selection of sounds in the entire soundscape; $\mathcal{D}$ is a distance function that returns a value based on the distance between the

current state and the goals state; $\mathcal{T}$ is a time penalty function that returns a value based on the time steps in one episode; and $(\mathcal{R}_d)_{d\in\mathcal{D}}$ is the reward that is calculated according to the distance function $\mathcal{D}$.

The state space is a k-dimensional discrete grid world and each dimension represents one sound in the soundscape. The volume is categorized to three levels, which are zero, low and high. The number of dimensions depends on how many sounds we have in the experiment.

The agent is only allowed to take action on one dimension at each step, where the action can be increasing or decreasing the volume by one level. There are $3^k$ states in the state space, and we assume there are no obstacles. That is, all combinations of sounds are possible. The size of the grid world is highly dependent on the number of dimensions, which might influence the algorithm's performance.

### 4.1.2 Tree Model

The Grid-World model assumes that all sounds in different categories have equal weight, which might result in a large number of redundant states. There is little chance that the goal is contained in redundant states as subjects prefer sounds under the same category in general. We introduce a tree model (see Figure 4.1) where a two-layer tree structure is applied for reducing the state space. In the first layer are category nodes; actual sound tracks are in the second layer, linked to the single category to which they belong.

We define the following tuple to represent the Tree model for this problem: $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, (\mathcal{R}_t)_{t\in\mathcal{T}} \rangle$ where $\mathcal{S}$ is a state space of two-layer tree; $\mathcal{A}$ is an action space where each action is the selection of sounds in the entire soundscape; $\mathcal{T}$ is a time function that returns a value based on the time steps in one episode; and $(\mathcal{R}_t)_{t\in\mathcal{T}}$ is the reward that is calculated according to time function $\mathcal{T}$.

Applying a hierarchical structure to the state space only allows sounds from the same category to be played together. Thus, the state space is reduced dramatically. The reward comes either from real subject experience or a virtual reward model. The real-world reward is measured by the length of time that a subject listens to a soundscape before pressing the "next" button (see Section

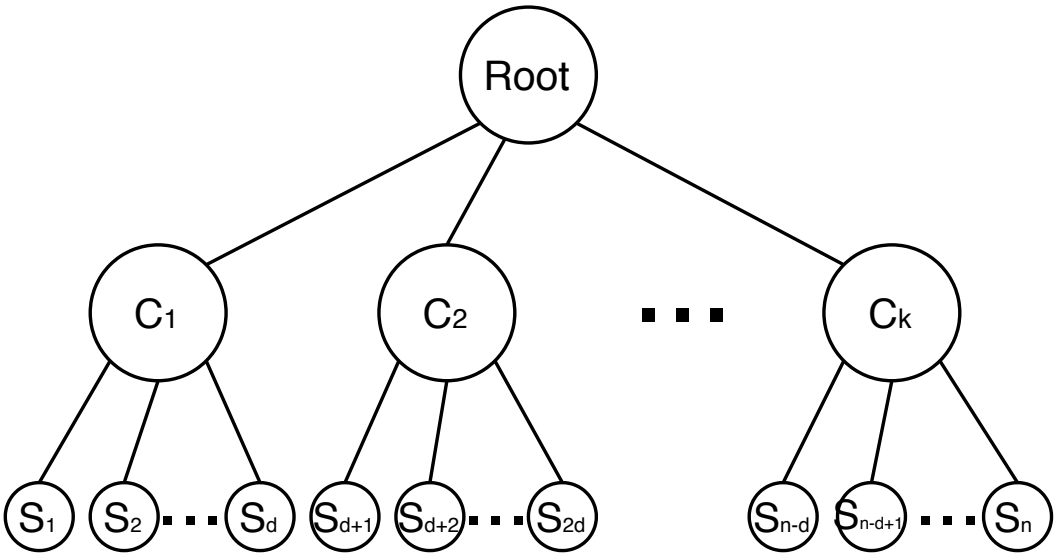



Figure 4.1: Tree model structure of the state space. The first layer consists of category nodes and the second layer contains all sounds.

4.2.1). Each node in the second layer has a ground truth value to represent the expected time length that the subject listens, and reward in each step is generated from a Gaussian distribution where the mean is its ground truth value.

## 4.2 Reward Model

In this study, we build a real-world reward model applied to human study experiments and two virtual reward models for experiments on virtual subject.

### 4.2.1 Real-world Reward

Real-world reward is utilized in the system that implements the one-button interface where listeners provide feedback by pressing the "next" button to switch a soundscape or the volume setting of currently-playing soundscape. Real-world reward is built on the one-button interface where it is measured by the elapsed time length $\mathcal{T}$ during which the real subject listens to the currently-playing soundscape. The reward is between 0 and 1: 1 if the listener does not click the button, and the reward is 0 if the listener clicks the button immediately after switching to a new soundscape. We define a reward func-

tion for listener's feedback where the time reward function is monotonically increasing since the longer the subject listens, the larger the reward is. We consider a sigmoid function that generate the reward $\mathcal{R}_{real-world}$[1]:

$$\mathcal{R}_{real-world} = \frac{1}{1 + e^{(-0.2\mathcal{T}+5)}}, \text{ where } \mathcal{T} \in [0, T_{max}]$$

### 4.2.2 Virtual Reward

A distance-based reward model and a one-button reward model are built for experiments on virtual subjects. Both reward models are based on Euclidean distance where the one-button reward model simulates the real-world reward better.

#### Distance-based Reward

Assume there exists a volume setting that has the best effect on increasing a subject's relaxation level, we can define a goal in the state space to indicate the best volume setting. Since some states that are far from the goal state might have the same negative effect, we introduce a radius $r$ to the distance function. The reward value is greater than 0 when the distance between current state and goal state is within the radius. Following this idea, we can define a distance-based reward function based on Euclidean distance $d$:

$$\mathcal{R}_{distance} = \begin{cases} \frac{1}{r} * (r - d), \text{ if } d < r \\ 0, \text{ if } d \geq r \end{cases}$$

#### One-button Reward

To simulate the real-world reward, we develop a one-button reward model for virtual subject experiments based on the Euclidean distance between the current-playing soundscape and the optimal soundscape in the state space.

[1]We selected the parameter values in the formula based on the assumptions: when $\mathcal{T} = 0, \mathcal{R}_{real-world} = 0$; and $\mathcal{T} = T_{max}, \mathcal{R}_{real-world} = 1$. We selected $T_{max} = 50$ in the experiments on real subjects.

We model the existence of an optimal volume setting of one soundscape that makes listener feel most relaxed. So, we can define a goal state in the state space to indicate the optimal volume setting. Since the real-world reward is measured by how long the subject decides to listen to the current soundscape, the parameter we simulate is the elapsed time $\mathcal{T}$ of playing one soundscape. The elapsed time $\mathcal{T}$ is dependent on the Euclidean distance $d$ between the current state and the goal state with additional noises generated from a Gaussian distribution $\mathcal{N}(0, 1)$.

$$\mathcal{T} = \frac{1}{d} \cdot T_{max} + \mathcal{N}(0, 1)$$

Similar to the real-world reward, the one-button reward $\mathcal{R}_{one-button}$ is generated from the simulated elapsed time $\mathcal{T}$ with sigmoid function:

$$\mathcal{R}_{one-button} = \frac{1}{1 + e^{(-0.2\mathcal{T}+5)}}, \mathcal{T} \in [0, T_{max}]$$

# Chapter 5

# Methods

In this work, we implement baseline algorithms, heuristic search algorithms, and bandit algorithm for experiments. Baseline algorithms are a random walk and linear search, which don't need to learn from the reward. Heuristic search requires prior knowledge of the environment along with an estimation of distance from the current state to goal.

## 5.1 Baseline Algorithms

Baseline algorithms are applied as the benchmark to compare and to assess the performance of proposed algorithms. We introduce random walk search and linear search in this work for comparison.

### 5.1.1 Random Walk Search

The agent starts with a random position and walks through the grid-world state space continuously where each step is to a random adjacent state.

### 5.1.2 Linear Search

The agent starts with a random position and sequentially runs through all the states in the grid world.

## 5.2 Real Time A*

Real Time A* (RTA*) is introduced by Korf [24] where the $h(n)$ is updated with the second best $f(n)$ in the open list when the agent moves to the next state. $f(n)$ is composed of $g(n)$ and $h(n)$ where $g(n)$ is the distance from the previous state to current state and $h(n)$ is an estimated cost of the path from current state to the goal state. The open list maintains a list of states that need to be visited. $h(n)$ is initialized to 0 at the beginning. $g(n)$ is independent of the initial state, which is appropriate for addressing our problem definition because the actual cost of a state only depends on the reward it receives from the environment. The basic idea of RTA* is to expand all neighbors around the current state and expand one neighbor with the minimum $f(n)$.

**Algorithm 3** RTA*

1: **while** $succ_i$ of s **do**
2: f($succ_i$) $\leftarrow g(s, succ_i) + h(succ_i)$
3: $h(s) \leftarrow$ second best $f(succ_i)$
4: move to succ$_i$ with min$f(succ_i)$

We simplified the state space model to make the agent work properly. In RTA*, each state has up to $3^k - 1$ neighbors in $k$-dimensional state space. The simplified agent only moves along a single axis at each step, so that each state has at most $2k$ neighbours. In addition, we modified the expansion path of the agent to avoid redundant moves. The RTA* agent goes back to the current state after expanding each neighbor. Instead of visiting the current state too many times, the simplified agent moves clockwise to expand all neighbors that are not visited before without returning to the current state.

## 5.3 UCT

Upper Confidence Bound applied to Trees (UCT) is an algorithm introduced by L. Kocsis and C. Szepesvari [2]. In UCT, each node is treated as an arm of a slot machine; thus this model becomes a multi-armed bandit model. UCT considers the bandit problem as a separate multim-armed bandit for each

node in the tree. It uses the UCB1 algorithm to select actions to maximize the reward at each node in the tree structure.

UCT is efficient at balancing exploration and exploitation with unknown branches in the tree search. The action selected to take at the current step is the node that has the maximum UCB value among all nodes in current layer, and every reward received in current node is averaged to the payoffs of previous node.

The idea of using this algorithm is that it helps to reduce the state space efficiently. The number of actions might increase dramatically when we increase the number of categories and soundscape examples. UCT allows us to identify if a large subset of actions is sub-optimal at an early stage, so we can improve performance by avoiding sampling redundant states multiple times.

**Algorithm 4** UCT algorithm [2]

```
1: function UCT
2:     while not Timeout do
3:         search(root, 0)
4: function SEARCH(node, depth)
5:     if Terminal(node) then return 0
6:     if Leaf(node) then Evaluate(node)
7:     nextNode ← argmax(nodes in depth+1)
8:     reward ← takeAction(nextNode)
9:     q ← updateValue(node, reward)
10:    return q
```

We assume that reward $X_{it} \in [0, 1]$, and the average reward value is $\bar{X}_{in} = \frac{1}{n}\sum_{t=0}^{n} X_{it}$. An action is selected according to:

$$I_t = \text{argmax}_{i\in[k]}\{\bar{X}_{i,T_i(t-1)} + c_{(t-1),(T_i(t-1))}\},$$
$$\text{where } c_{t,s} = 2C_p\sqrt{\frac{\ln t}{s}}.$$

$I_t \in [1, ..., k]$ is the index of arm selected in time step t. $T_i(t)$ is the number of times arm i was played up to time t (including time t).

# Chapter 6

# Experiments on the Virtual Subject

The experiments on the virtual subject are tested on both a distance-based reward model and a one-button reward model. We evaluated our proposed system in two different ways. We compared both the distance between the goal and the last state the agent reaches and the discounted cumulative rewards collected along the path given a fixed number of time steps. Unlike the algorithms running on grid world, UCT has a different strategy of exploring and exploiting. More precisely, the distance evaluation on UCT is based on the distance between the goal state and the current state with the highest mean of the state value.

The selection of the start state could impact the experiment results significantly. With the assumption that the state space has a potential hierarchical structure, we define the goal to be a combination of volume levels of sounds where all sounds are under one of the soundscape categories. In other words, the goal is a soundscape category at any volume levels (low or high) instead of a mix of sounds in different categories. The agent running RTA* algorithm might have different performance with a different location of start state because there exist many zeros (silent sounds) in the goal state. Thus, we tested on both cases where the start state locates at the origin of the Euclidean state space (all sounds are silent), or the start state is a combination of random volume levels of randomly selected sounds.

The virtual experiment has 20 trials and each trial has 1000 runs. A run is

an episode where the agent completes a task and reaches the goal. There is a fixed number of time steps for each trial ranging from 10 to 200 steps sampled at 10-step intervals (e.g. 10, 20, ... 190, 200). We compared Real-Time A* and UCT with two baseline algorithms (linear search and random walk search) in all experiments.

## 6.1 One-button model

The one-button reward model generates the reward that is dependent on the Euclidean distance. It simulates the elapsed time between button presses based on Euclidean distance and uses the sigmoid function for generating the reward from the time (See Section 4.2.2). We conducted experiments on one-button reward model and measured with the distance and the discounted cumulative rewards. We present two Figures 6.1 and 6.2 in this section to display the evaluation on the mean of distance and discounted cumulative rewards with 95% confidence intervals.

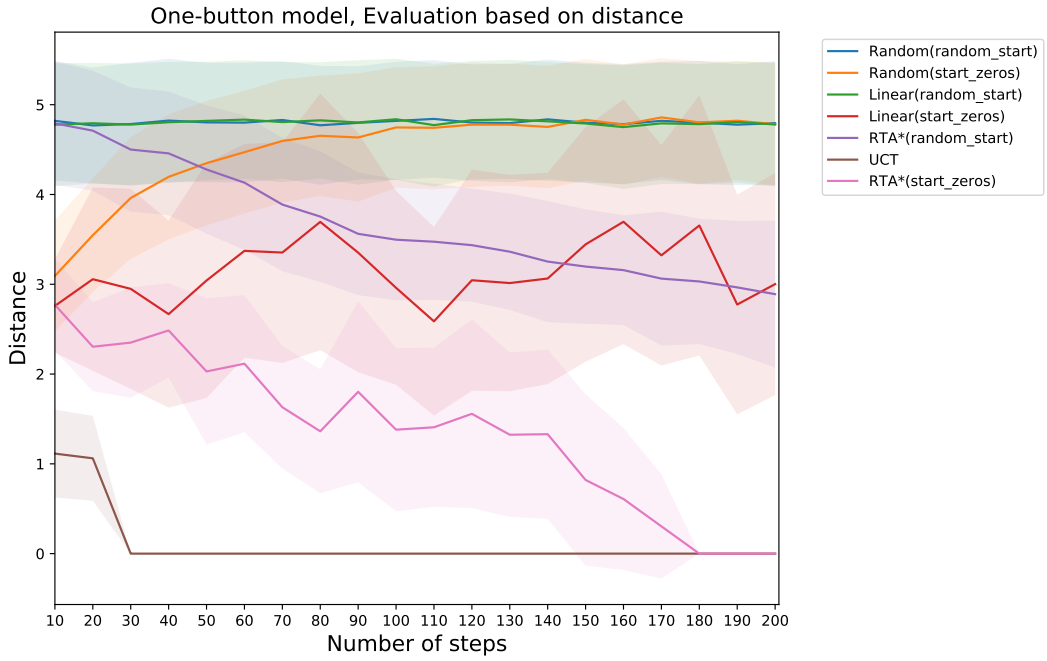



Figure 6.1: Distance over time of baselines random search, linear search, and UCT, and RTA*. Low distance is optimal. The shadings are 95% confidence intervals for each algorithm.

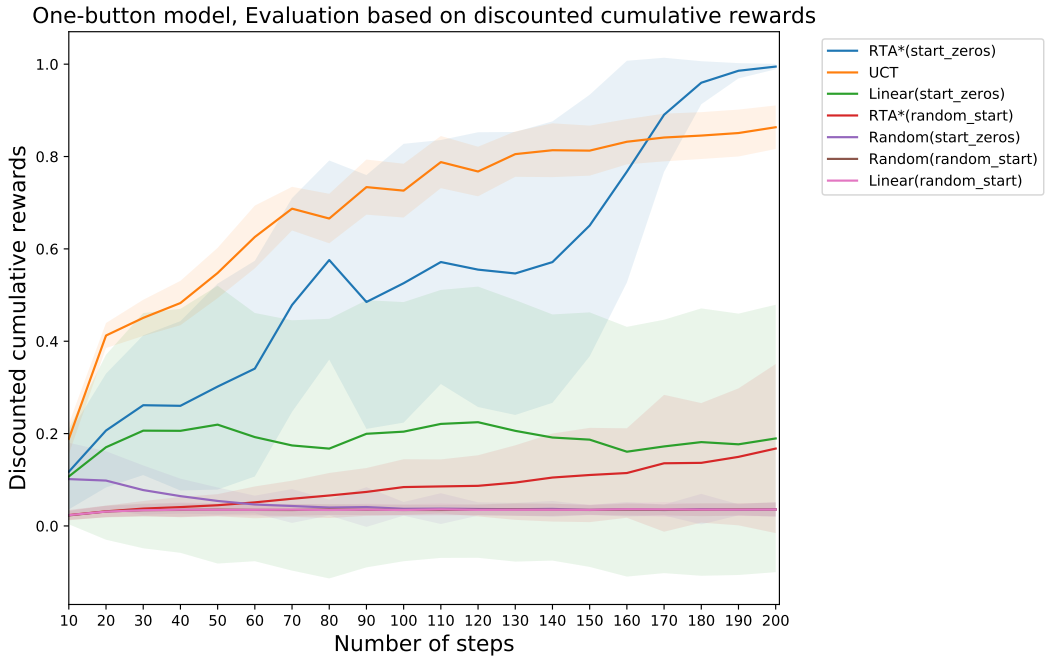



Figure 6.2: Discounted cumulative reward of baselines, UCT, and RTA*. Higher reward sooner indicates better performance. The shadings are 95% confidence intervals for each algorithm.

Figure 6.1 reports the virtual subject experiment evaluated by distance. Decreased distance means that the current state is moving closer to the goal state. The curves are not smooth because the reward can be different even if the distance between the current state and the goal state is the same, since noise is being added to the Euclidean distance and sigmoid functions are applied to the simulated elapsed time. UCT requires less time to distinguish the optimal and sub-optimal nodes when the true mean values of optimal and sub-optimal are significantly different.

Figure 6.2 shows the virtual subject experiment evaluated by discounted cumulative rewards. From this figure we can tell that UCT performs better than RTA* before 160 steps, but worse afterwards because it continues to sample. UCT is efficient at determining the optimal state within a small number of steps, despite the fact that discounted cumulative reward can be affected by exploring other states continuously. According to the result of the virtual subject experiment, RTA* is also able to discover the goal state eventually.

For the one-button reward model, UCT outperforms RTA* when we evaluate on distance. RTA* is able to determine the goal within 200 steps when the start state is the origin of the Euclidean space, and it outperforms UCT after around 160 steps because RTA* keeps sampling sub-optimal actions. Both of UCT and RTA* display better performance than the two baseline algorithms.

## 6.2 Distance-based model

The distance-based reward model generates the reward based on the distance between the agent and the goal state (See Section 4.2.2). The distance-based reward model is tested and measured with the distance and the discounted cumulative rewards. We display Figures 6.3 and 6.4 in this section to illustrate the evaluation on the mean of distance and discounted cumulative rewards with 95% confidence intervals.

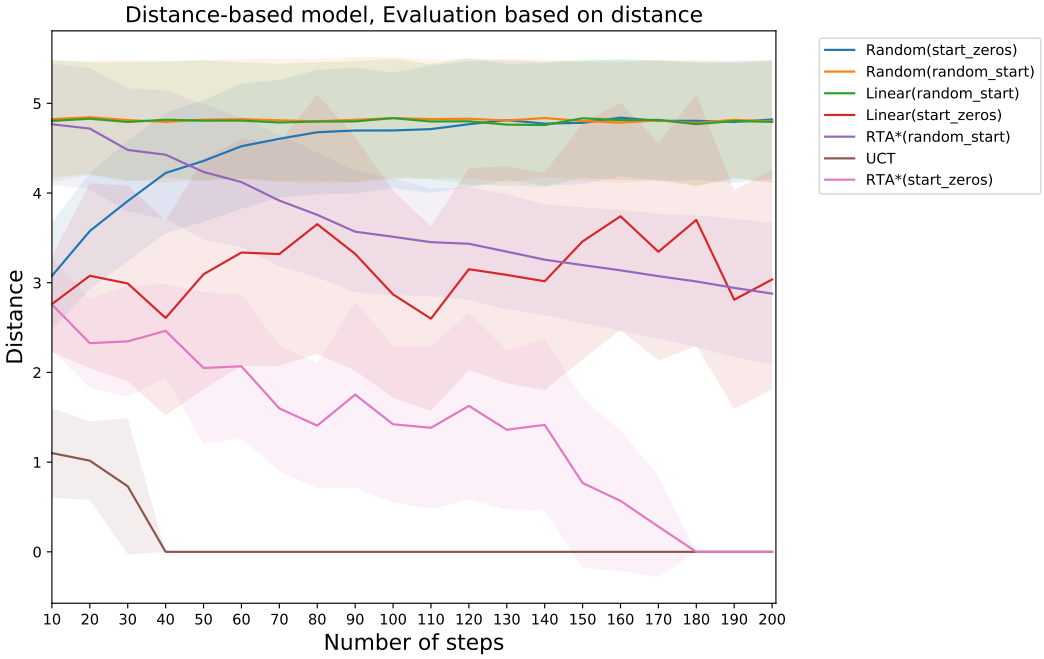



Figure 6.3: Distance over time of baselines random search, linear search, and UCT, and RTA* with start state being random and all zeros. Low distance is optimal. The shadings are 95% confidence intervals for each algorithm.

Figure 6.3 shows the experiment on distance-based reward model measured by distance. UCT is able to determine the state with maximum mean after

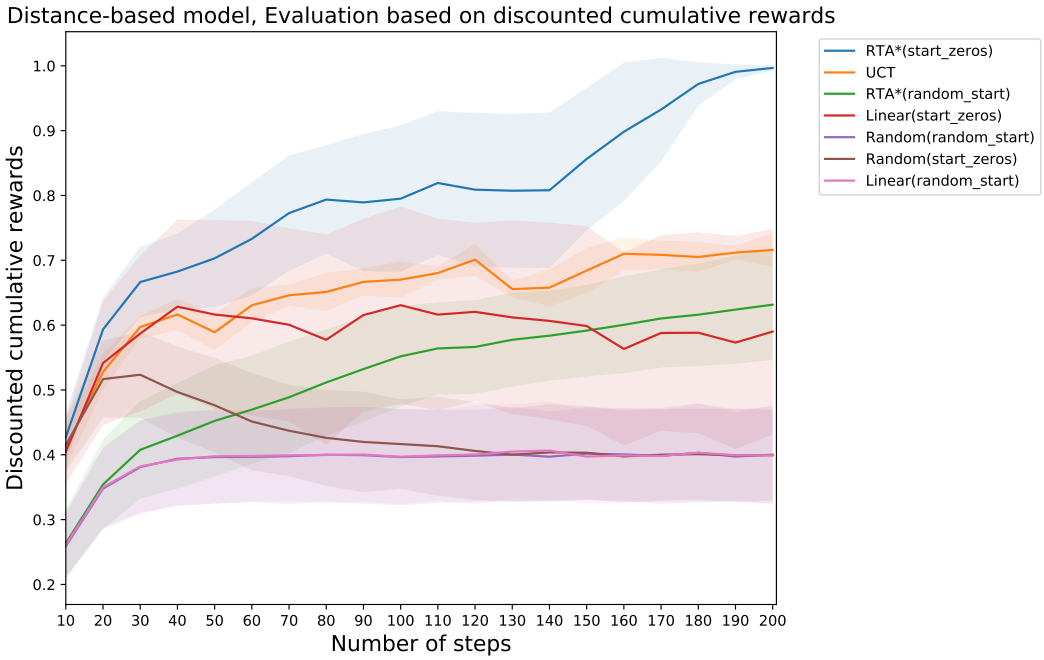



Figure 6.4: Discounted cumulative reward of baselines, UCT, and RTA* with start state being random and all zeros. Higher reward sooner indicates better performance. The shadings are 95% confidence intervals for each algorithm.

about 40 steps, and RTA* with start state being all zeros spends around 180 steps to find the goal. When the start state is randomly selected, the distance from the start state to the goal state can be relatively large. Thus, there is a big gap between UCT and other algorithms at 10 steps as UCT can quickly find the node in the first layer with maximum value. The curve of RTA* declines as number of steps increases, but 200 steps are insufficient for finding out the goal state.

Figure 6.4 shows the experiment on distance-based reward model measured by discounted cumulative rewards. RTA* with start state being zeros performs better than UCT when measured on discounted cumulative rewards. Both UCT and RTA* increases when number of steps increases, and from this figure we know that UCT and RTA* can't converge within 200 steps. Although it spends 40 steps to find out the state with the maximum mean, it needs more than 1000 steps of sampling to determine the true value of each state (UCT can barely converge after 1000 steps). In Figure 6.4, UCT has a small drop at 50 and 120 steps which is caused by sampling sub-optimal actions.

Similar to the results in Section 6.1, UCT running on distance-based reward model outperforms RTA* when measured on distance. RTA* with start state being all zeros outperforms UCT when we evaluate them on discounted cumulative rewards. Compared to one-button reward model, UCT has a worse performance while RTA* has a better performance when running on distance-based reward model.

## 6.3 Summary

As the one-button reward model is dependent on Euclidean distance, algorithms running on grid-world environments based on a one-button reward model perform similar to the distance-based reward model. We can conclude that UCT has a better performance on the one-button model compared to distance-based model as UCT is able to determine the soundscape with the maximum mean value within 30 steps, which is 10 steps less than the result of a distance-based model. In the distance-based reward model, all child nodes that do not share the goal node could have the same Euclidean distance to the goal state. In the one-button reward model, the reward is also dependent on distance but each node has a different reward because of noise being added to the Euclidean distance and sigmoid function being applied to the simulated elapsed time. UCT performs well when the differences of nodes are larger. Thus, UCT outperforms all grid-world algorithms within a limited number of time steps.

# Chapter 7

# Experiments on Real Subjects

In the previous chapter, we presented results of testing all algorithms on virtual subjects. In this chapter, we report results of experiments on real subjects.

## 7.1 Experimental Setup

Subjects were asked to participate in a 20-minute experiment. They interacted with the system that implements one-button interface by pressing the "next" button to switch to the next soundscape or another volume combination of the current soundscape. Subjects were asked to press the "next" button whenever they felt the currently playing soundscape was not relaxing. Each soundscape could be played up to 50 seconds with a cross-fade before switching to the next soundscape.

In the soundscape there are 15 nature sounds in 5 different categories, where each category has 3 sounds. The categories are forest, ocean, night-time camping, rain on a tent, and city rain. Each sound has 3 different levels of volume: silent, low and high. Subjects are always exposed to each of the 5 categories in order with all 3 sounds playing at "low" before repeating a category and/or adjusting the volume. After the experiment, subjects were required to fill out a survey to report on their experiences and provide feedback on the session.

## 7.2 Recruitment

An experimental evaluation on 22 real subjects was conducted. The subjects were undergraduate students with different majors, graduate students with different majors, and professors[1].

Subjects were divided into two groups: the treatment group and the control group. Subjects in treatment group listened to soundscapes selected by the UCT algorithm and initial soundscapes were also selected by UCT algorithm. The reason we used the UCT algorithm in this experiment is that this algorithm performed better than RTA* in the virtual experiments (see Chapter 6). Subjects in the control group listened to soundscapes selected by the experimenter, and the initial soundscapes were selected in order. The selection strategy was based on how long the subject had listened to the currently-playing soundscape. If the subject switched a soundscape immediately, this soundscape would not be selected for a long time until the rest of soundscapes were recognized as not relaxing by the experimenter. If the subject did not press the "next" button for a particular soundscape, various combinations of volume levels would be chosen. Each soundscape might be played multiple times. Pressing the "next" button either switched to a different soundscape or changed the volume of sounds in the current soundscape.

## 7.3 Execution

All subjects were asked to sit in a chair or lie on a bed to ensure they felt comfortable with the surroundings. Online experiments were conducted using the AnyDesk application to control the experiment machine remotely. All subjects were asked to check the audio settings in their own machine after connecting to the experiment machine to make sure they could hear sounds from either their speaker or headphones. Subjects assumed they listened to sounds selected by the automated sound therapy system. They knew that there were 5 categories of natural sounds in the experiment and the system

[1]Due to the limitation caused by COVID-19 crisis, 18 out of 22 tests were done online and 4 were done in person.

would select soundscapes and fine-tune their component sounds. The subjects did not know the experimenter's sound selection strategy or the algorithm used in the system; thus they were not aware what sounds would be played after pressing the "next" button.

Subjects could press the ENTER key to switch soundscapes during the experiment. If subjects felt the sound they were currently listening was not pleasing or did not make them feel relaxed, they could switch to the next sound by pressing the ENTER key. If they felt the sound they were currently listening to was pleasing, they could keep listening to it. Subjects were asked to fill out a survey form after the experiment to provide the feedback of their experiences about the experiment.

## 7.4 Results

In this section, we present the results from different perspectives such as the summary of the experience, Subject behaviors and correlations among button-press frequency, rewards and user ratings.

### 7.4.1 Subjects' Experiences

Based on the survey responses, 14 of 22 felt that the system eventually discovered the sound they felt to be most relaxing. Figure 7.1 demonstrates the distribution of the ratings of how relaxed subjects felt about the last soundscape chosen by the system. The upper plot is for the treatment group and the lower plot is for the control group. The user ratings range from 1 to 10 where the higher rating indicates that the system is more effective at determining the relaxing soundscape and vice versa. We can conclude from this figure that most users reported the system was able to choose relaxing sounds before the end of the experiments no matter whether sounds are selected by UCT algorithm or experimenter.

We compared the user ratings of two groups using the Mann Whitney U test. The sample size of the treatment group is 15, and the sample size of the control group is 7. We run the test at a 5% level of significance (i.e., $\alpha$=0.05)

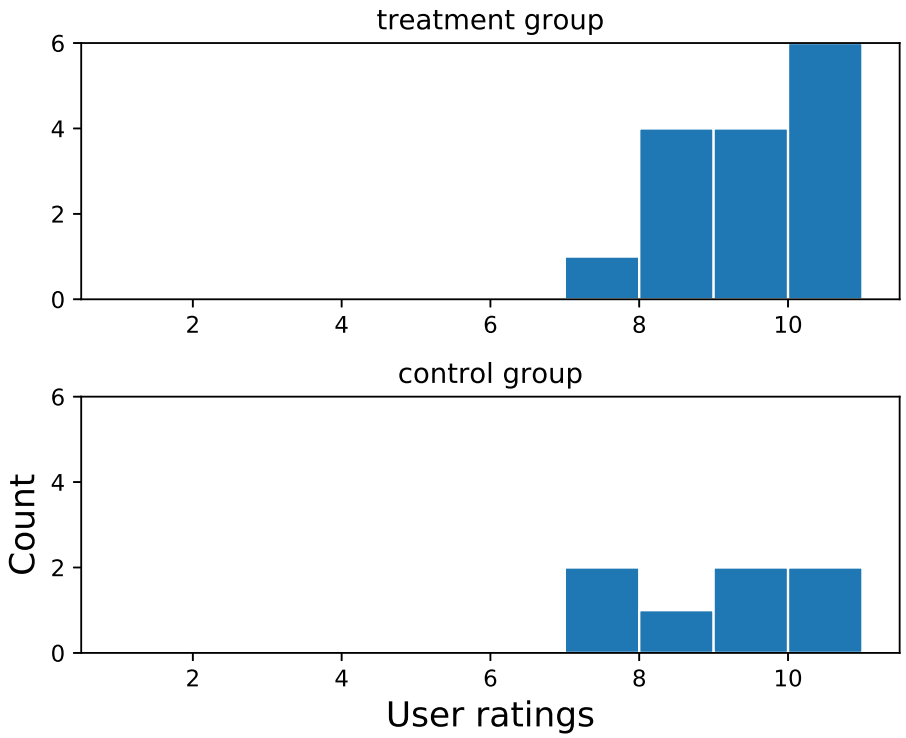



Figure 7.1: Survey results for the question: "Did the system eventually choose the most relaxing sound?". Distribution of the ratings of whether the system eventually chose the most relaxing sound. Here, 10 is excellent performance, while 1 is unsatisfactory performance.

given the hypothesis:

$H_0$: The two distributions are equal, versus

$H_1$: The two distributions are not equal.

The U value is 42.0, and according to the table of critical values of U we do not reject the null hypothesis $H_0$ because $42.0 > 24.0$. We do not have sufficient evidence to conclude that the treatment group differs from control group in terms of user ratings. Even though there is no significant difference between the UCT algorithm and the experimenter, it is sufficient to imply that experimenters could be replaced or automated by using the UCT algorithm.

To better understand the diversity of listeners' preferences on sound categories, Figure 7.2 shows the distribution of all subjects' self-reported most

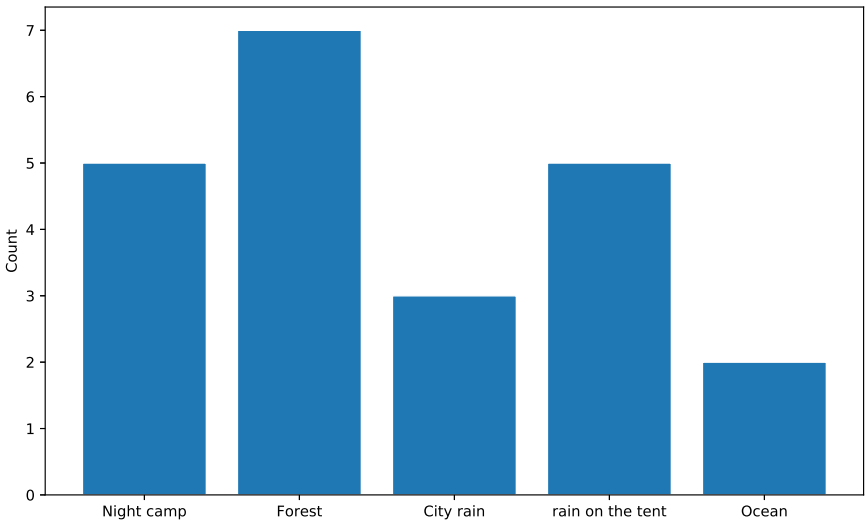



Figure 7.2: Distribution of the most relaxing sound from the surveys

relaxing sounds from the surveys. 35% of subjects preferred the sound of forest, and 20%-25% of subjects felt most relaxed listening to one of night-time camping, city rain or rain on a tent. The selection of sounds and sound categories may influence the sound therapy session significantly. For example, a sound in an arbitrary sound category might affect the subjects' responses towards the whole category if the sound is not relaxing.

### 7.4.2 Subject Behaviors

We collected the subjects' interactions with the system during the experiments. Based on the subject interactions, we analyze behaviors such as how often the subjects press the "next" button, and how quickly they press the "next" button when they feel the soundscape they are listening to is not relaxing.

Figure 7.3 displays the reward given by subjects in treatment group over time. Figure 7.4 shows the reward given by subjects in control group over time. In both figures, the maximum reward is represented by 1 and the minimum reward is represented by 0. The participant indices presented in the figure are sorted increasingly by cumulative rewards. We expected those who didn't press the "next" button too often had a large value of cumulative rewards.

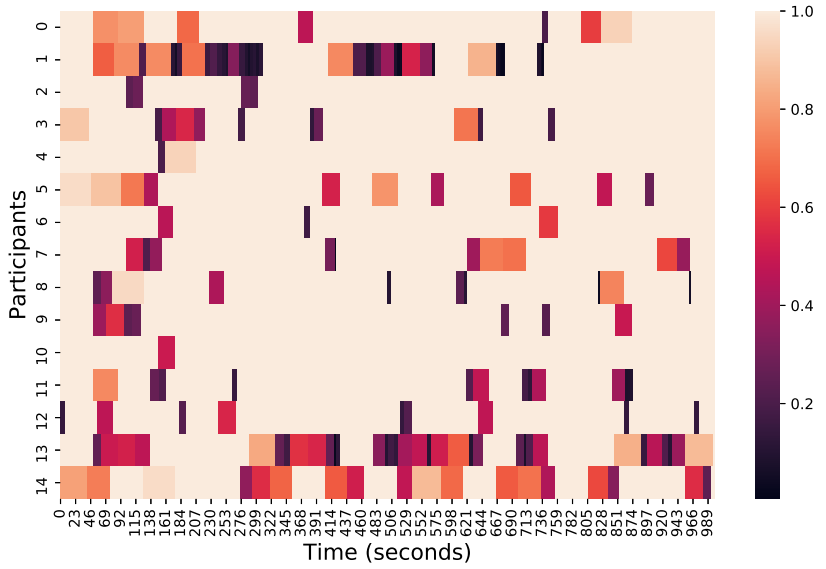



Figure 7.3: Depiction of reward over time for each participant in treatment group (0th to 14th). Dark bars indicate low reward (clicking the "next" button sooner), and light indicates high reward (clicking later or not at all). The dominance of light colour indicates listeners were either satisfied most of the time or our granularity of scoring reward is too fine or too quick.

But both of the plots indicate that those who pressed the "next" button many times might have a larger value of cumulative rewards. This is caused by the reward model function where the sum of rewards received in a period of time might be greater when the subject presses the button many times comparing to doing nothing. Most of the subjects reported a positive experience with the experiments, according to Figure 7.3 and 7.4. In general, the decrease of button-press frequency indicates that the real subject is getting more and more satisfied with the current soundscapes.

Moreover, we attempt to observe the subjects' behaviors and analyze the effectiveness of the reward function. Figure 7.5 and Table 7.1 demonstrate the distribution and summary statistics of the duration between the pressing of the "next" button for treatment group. Generally, the number of "next" button presses decreases when the elapsed time increases as the median and the third quartile of duration between pressing the "next" button were 17.34 and 26.32 seconds. Most of the button-press actions occurred before 26.32 seconds

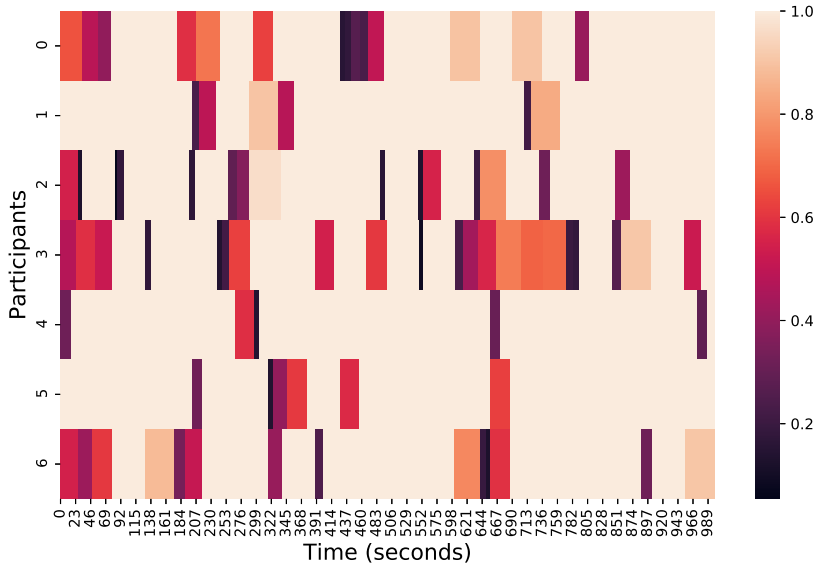



Figure 7.4: Depiction of reward over time for each participant in control group (0th to 6th). Dark bars indicate low reward (clicking the "next" button sooner), and light indicates high reward (clicking later or not at all). The dominance of light colour indicates listeners were either satisfied most of the time or our granularity of scoring reward is too fine or too quick.

when subjects didn't feel relaxed. Some self-reported feedback suggested that subjects can get bored and change the category preferences over time. Thus we can interpret that the reward model built on elapsed time is effective and practical.

| Min | 25% | Median | Mean | 75% | Max | Std. |
|---|---|---|---|---|---|---|
| 0.00 | 5.77 | 13.07 | 17.34 | 26.32 | 48.03 | 13.16 |

Table 7.1: Summary statistics of duration between pressing the next button for the treatment group

Figure 7.6 and Table 7.2 demonstrate the distribution and summary statistics of the duration between the pressing of the "next" button for control group. Generally, the number of "next" button presses decreases when the elapsed time increases as the median and the third quartile of duration between pressing the "next" button were 21.60 and 22.59 seconds. From Figure 7.5 and 7.6 we observe that the two groups had different behaviors in terms of pressing the "next" button. This might be affected by the smaller number of

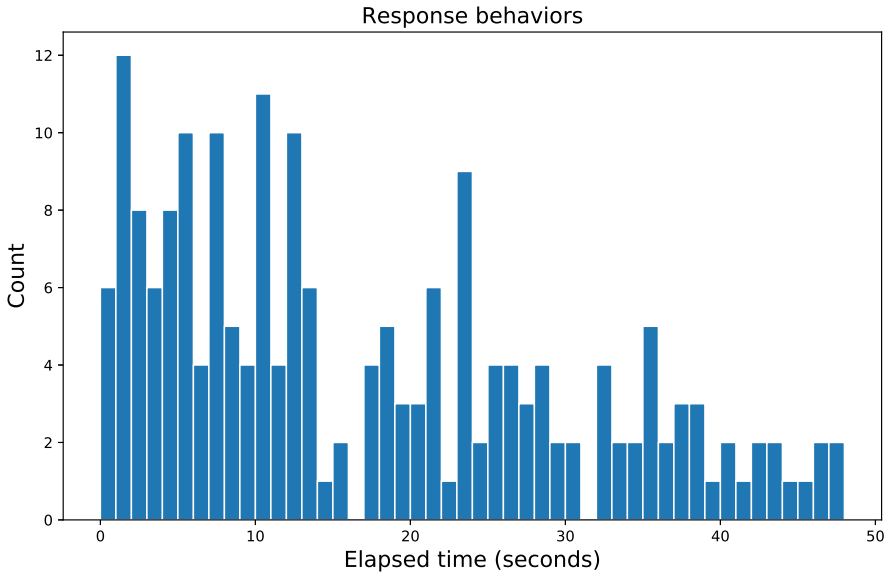



Figure 7.5: Distribution of duration between pressing the next button. The median was 13.07 seconds, the average was 17.34 seconds, and the standard deviation was 13.16 seconds.

subjects in control group, or a different strategy of sound selection.

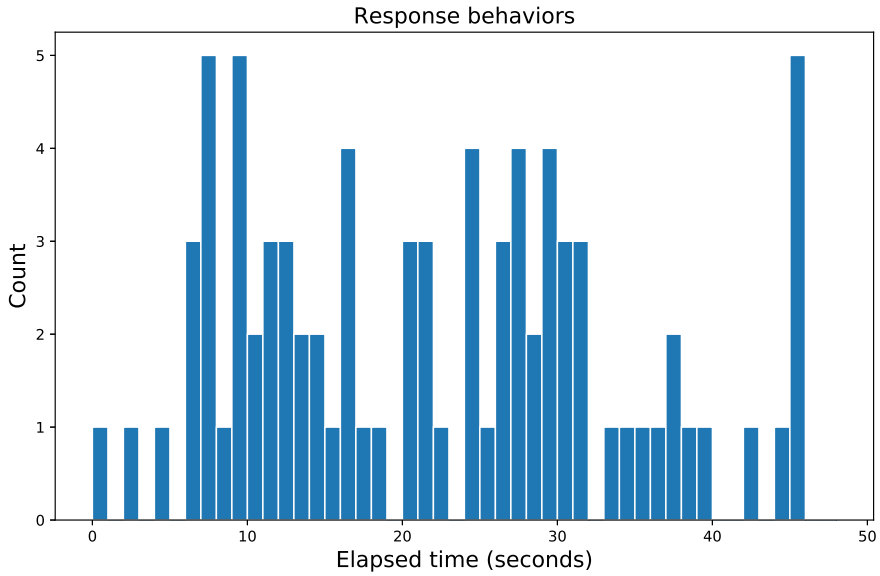



Figure 7.6: Distribution of duration between pressing the next button. The median was 21.60 seconds, the average was 22.59 seconds, and the standard deviation was 12.19 seconds.

| Min | 25% | Median | Mean | 75% | Max | Std. |
|---|---|---|---|---|---|---|
| 0.46 | 11.99 | 21.60 | 22.59 | 30.59 | 48.52 | 12.19 |

Table 7.2: Summary statistics of duration between pressing the next button for the control group

### 7.4.3 Factor Correlations

In order to further evaluate the system and have a better understanding of how it performs, we measure the correlations among button-press frequency, user ratings and cumulative rewards. Figure 7.7, 7.8, and 7.9 give an overview of the relations between button-press frequency and user ratings, button-press frequency and cumulative rewards, and cumulative rewards and user ratings.

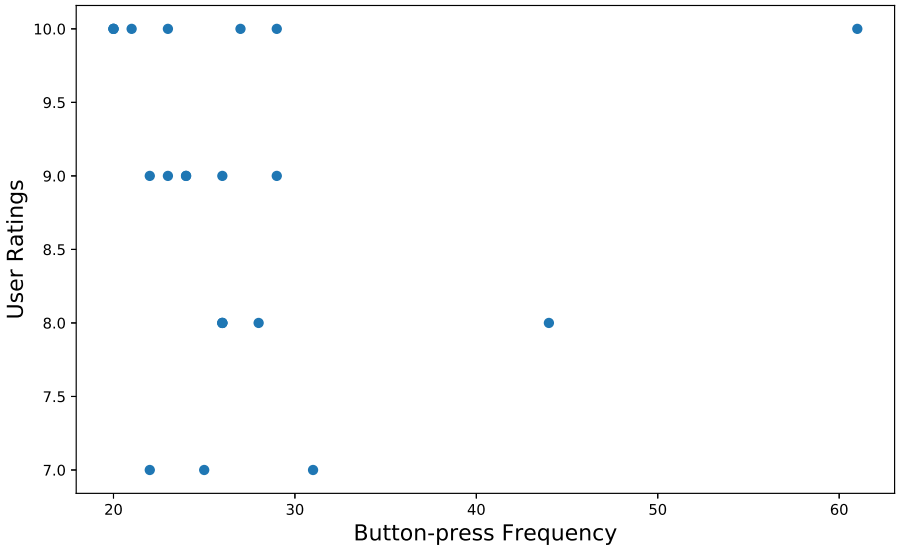



Figure 7.7: Button-press frequency vs. User ratings

The Pearson correlation coefficient between button-press frequency and user ratings is -0.02 with p-value being 0.94, as presented in Figure 7.7. We expect to see that lower button-press frequencies have higher user ratings. But the result indicates that the button-press frequency and user ratings are uncorrelated which means higher button-press frequency does not imply a worse sound therapy experience.

There is no linear correlation between button-press frequency and cumu-

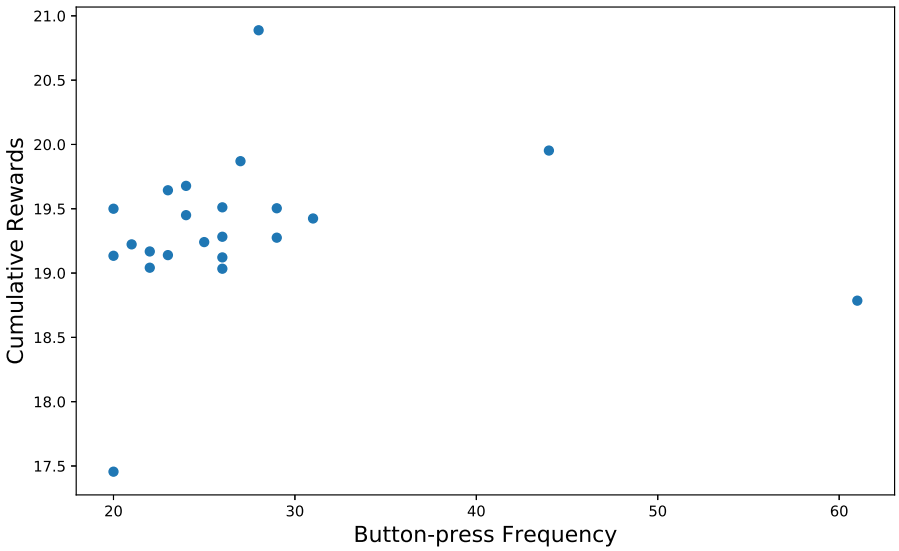



Figure 7.8: Button-press frequency vs. Cumulative rewards

lative rewards as the Pearson correlation coefficient is 0.08 and its p-value is 0.73. Figure 7.8 reflects the weakness of the reward function where the cumulative rewards in a fixed time duration might be higher when the button-press frequency is higher. If we have a proper reward model, the cumulative rewards can be higher if the button-press frequency is lower. In the real-world reward model, we use a sigmoid function to generate the reward from the elapsed time between button presses. The sigmoid function increases slowly at the beginning and the end, and increases fast in the middle. So, the sum of rewards with multiple button presses in a time period might be higher than the maximum reward in a time period (without a button press).

We expected to see the user rating positively correlate with cumulative rewards, but Figure 7.9 indicates that the user rating has no significant relation with cumulative rewards. The Pearson correlation coefficient between these two variables is -0.21 with p-value being 0.36.

Those figures demonstrate that the reward function might not be able to represent listener's preferences well, or the measurement of evaluation cannot reflect the listener's experience properly.

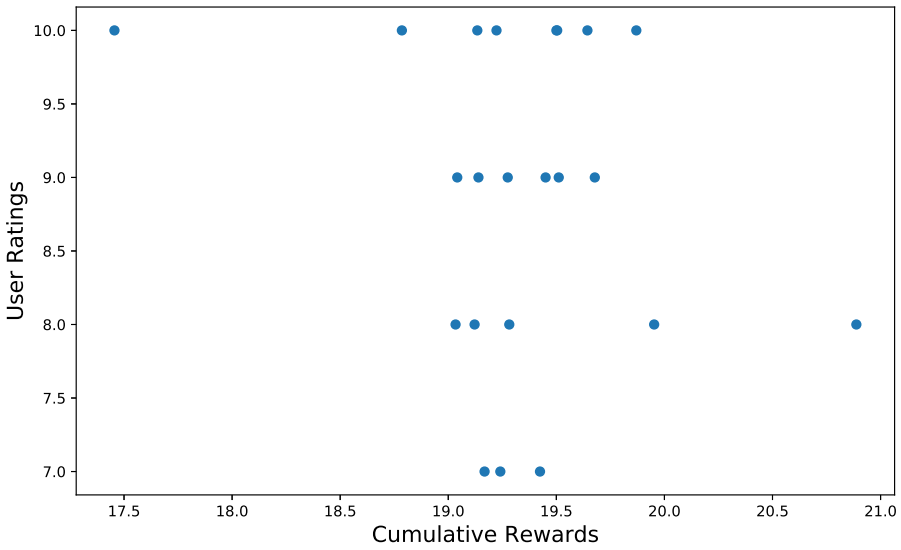



Figure 7.9: Cumulative rewards vs. User ratings

## 7.5 Threats to Validity of the Experiment

In this section, we discuss threats to validity of the experiments. We analyze some factors that might affect the internal validity, external validity and construct validity of experimental results.

### 7.5.1 Interval Validity

Subjects were not randomly assigned to the treatment group and the control group. We assigned first 15 subjects to the treatment group and the rest of 7 subjects to the control group. The results can be biased when subjects are not randomly assigned.

### 7.5.2 External Validity

The population of the subjects was mostly university students and professors with different specializations. Participants are a particularly well-educated segment of society, and that most of them were young. It might impact the result when we apply it to the general population.

### 7.5.3 Construct Validity

The experiments were performed remotely because of COVID-19 pandemic, and it affected the experiment settings such as the surroundings around the subject, whether the subject uses headphones/earbuds or speaker. Furthermore, measurement on the influence of volume is limited as the subjects' perception levels of volume vary. This factor may bias the correlation of the user ratings and system performance. Since some of the subjects can barely distinguish the difference of volume and the sound selection is based on the subjects' feedback of currently listening sound, their feedback might prevent the system from making good decisions. In addition, the experimenter is an amateur of sound therapy, and a well-trained sound therapist might outperform the UCT algorithm.

From the self-reported responses we also observe that the number of categories might be too small for exploration in the experiment. User behaviors were different with regard to pressing the "next" button. For example, from many optional interviews we learn that some preferred listening to the favourite soundscape all the time while some preferred to switch soundscapes all the time. Most of the subjects preferred listening to more than one soundscapes during the experiment. A small number of subjects changed preferences over time because of getting bored with a particular soundscape.

The mental health information of participants such as sleep quality, stress and anxiety level were unknown before the experiment for either the treatment group or the control group. We didn't compare the stress and anxiety level before and after the experiments for both the treatment group and the control group.

## 7.6 Reflections on Experiments

Other than the threats to experimental validity mentioned in the previous section, some issues remain that could be addressed and fixed in future experimental research. In terms of the experiments design, we did not compare the stress level of subjects before and after the experiment. Without the compar-

ison, we were able to display the correlation between our work and positive outcomes, but we can not show direct causation because subjects might be already relaxed before the experiments. Apart from the comparison between the stress level before and after the experiments, it would be better if we could add a stressor to the subject before the experiment. Thus, State-Trait Anxiety Inventory (STAI) [32] could be applied as a psychological measure for applying a stressor and evaluating the subject's stress level. We should ensure that the subjects are in the same stress level before the experiments so that the stress level does not bias the result.

Due to the limitation caused by COVID-19, we conducted the experiments remotely. If we were able to perform the in-person experiments, we could use physiological measures such as blood pressure, heart rate, and heart rate variability (HRV) to evaluate the stress level before and after the experiments. After the experiments, we should be able to collect both physiological and psychological data from the experimental group and the control group. We could apply a paired Mann Whitney U test to show if there was a significant effect.

With regard to the survey, the answers for the previous three questions scale from 1 to 10 (See Appendix E). Questions are:

1. Do you feel the algorithm eventually chose the soundscape that makes you feel most relaxed?

2. Can you estimate how long it took to determine the soundscape that makes you feel most relaxed?

3. Do you like all soundscapes chosen by the program?

The forth question ("How fast did you respond your feedback to the current playing soundscape?") had another scale from 1 to 5, so we need to unify the scales. Since subjects might be confused by different scales, using a consistent Likert scale [27] with fewer responses could improve the consistency of survey responses.

## 7.7 Summary

Based on the surveys and subjects' feedback, we conclude that UCT performed well at determining the listeners' sound preferences in a small number of steps. Although the control group exhibited no significant difference compared to the treatment group, we can infer that the UCT algorithm is able to replace the experimenter to conduct sound therapy.

Yet the human study can be affected by many factors apart from the algorithm such as the diversity of sound categories, the quality of sounds and the change of personal preferences over time. So, further experiments are needed to improve validity.

# Chapter 8

# Conclusion

In this work we present an interactive learning system that automatically selects and fine-tunes soundscapes according to personal preferences obtained from minimal user feedback intended to help relax the subjects. We conducted experiments on both virtual subjects and real subjects. For experiments on virtual subjects, we tested UCT, RTA* and baseline algorithms on both a one-button reward model and a distance-based reward model. UCT had the best performance among all algorithms in terms of the evaluation on distance and discounted cumulative rewards within a limited number of time steps. Based on the results of experiments on virtual subjects, we tested UCT algorithm with one-button reward model on real subjects. Subjects were divided into a treatment group and a control groups for comparison, where the treatment group listened to sounds selected by the UCT algorithm and the control group listened to sounds selected by the experimenter. The UCT algorithm performed well in determining the most relaxing soundscape. While the control group exhibited no difference compared to the treatment group, we can still conclude that the UCT algorithm is able to replace the experimenter for sound therapy.

There are several directions to pursue in future investigations. Since personal preferences might change over time during the experiment, we could model preference as a non-stationary problem, and apply a non-stationary bandit algorithm. In addition, we could increase the number of sounds and sound categories in the human study to better illustrate the algorithm's ef-

fectiveness. Increasing the diversity of soundscape categories may provide a wider range of choices and create a more even distribution of preferences to eliminate the distribution bias. Another reward model might be developed to ensure that the reward is not too time-sensitive. Our ultimate goal of the work is to apply this system to help hospitalized patients who suffer from stress and anxiety caused by the environment noises after surgeries and treatments. Beyond sound therapy, these results could be reused for music recommendation systems.

# References


[1] K. Allen, L.H. Golden, M.I. Ching, A. Forrest, C. Niles, P. Niswander, J. Barlow, and Jr. Izzo, J.L. Normalization of hypertensive responses during ambulatory surgical stress by perioperative music. *American Journal of Hypertension*, 11(S3), 1998.

[2] Jean-Yves Audibert, Remi Munos, and Csaba Szepesvári. Variance estimates and exploration function in multi-armed bandit. 2007.

[3] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47, 2002.

[4] David A. Barger. The Effects of Music and Verbal Suggestion on Heart Rate and Self-Reports. *Journal of Music Therapy*, 16(4), 1979.

[5] V. Bulitko and G. Lee. Learning in real-time search: A unifying framework. *Journal of Artificial Intelligence Research*, 25, 2006.

[6] Vadim Bulitko. Learning for adaptive real-time search, 2004.

[7] Jarad Cannon, Kevin Rose, and Wheeler Ruml. Real-time motion planning with dynamic obstacles. In *SOCS*, 2012.

[8] Shuo Chen, Josh L. Moore, Douglas Turnbull, and Thorsten Joachims. Playlist prediction via metric embedding. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, New York, NY, USA, 2012. Association for Computing Machinery.

[9] Cynthia M. Corhan and Beverley Roberts Gounard. Types of music, schedules of background stimulation, and visual vigilance performance. *Perceptual and Motor Skills*, 42(2), 1976. PMID: 1272711.

[10] Diane Snyder Cowan. Music Therapy in the Surgical Arena. *Music Therapy Perspectives*, 9(1), 1991.

[11] C. Dhahri, K. Matsumoto, and K. Hoashi. Mood-aware music recommendation via adaptive song embedding. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2018.

[12] Rüdiger Ebendt and Rolf Drechsler. Weighted a search – unifying view and application. *Artificial Intelligence*, 173(14), 2009.

[13] David Evans. The effectiveness of music as an intervention for hospital patients: a systematic review. *Journal of Advanced Nursing*, 37(1), 2002.

[14] Darryl Griffiths, Stuart Cunningham, and Jonathan Weinel. A discussion of musical features for automatic music playlist generation using affective technologies. 2013.

[15] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2), 1968.

[16] Jason K. Howlett, Timothy W. McLain, and Michael A. Goodrich. Learning real-time a* path planner for unmanned air vehicle target sensing. *Journal of Aerospace Computing, Information, and Communication*, 3(3), 2006.

[17] Makoto Iwanaga and Youko Moroki. Subjective and Physiological Responses to Music Stimuli Controlled Over Activity and Preference. *Journal of Music Therapy*, 36(1), 1999.

[18] Isuru Jayarathne, Cohen Michael, Frishkopf Michael, and Mulyk Gregory. Relaxation "sweet spot" exploration in pantophonic musical soundscape using reinforcement learning. *In Proceedings of the 24th International Conference on Intelligent User Interfaces: Companion, 55–56. IUI '19. New York, NY, USA: ACM*, 2019.

[19] Emilie Kaufmann, Olivier Cappe, and Aurelien Garivier. On bayesian upper confidence bounds for bandit problems. In Neil D. Lawrence and Mark Girolami, editors, *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, volume 22 of *Proceedings of Machine Learning Research*, La Palma, Canary Islands, 2012. PMLR.

[20] In-Cheol Kim. Exploring an unknown environment with an intelligent virtual agent. In Jérôme Euzenat and John Domingue, editors, *Artificial Intelligence: Methodology, Systems, and Applications*, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[21] Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou, editors, *Machine Learning: ECML 2006*, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[22] Sven Koenig and Xiaoxun Sun. Comparing real-time and incremental heuristic search for real-time situated agents. *Autonomous Agents and Multi-Agent Systems*, 18, 2009.

[23] Richard E. Korf. Depth-first iterative-deepening: An optimal admissible tree search. *Artificial Intelligence*, 27(1), 1985.

[24] Richard E. Korf. Real-time heuristic search. *Artificial Intelligence*, 42(2), 1990.

[25] Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.

[26] Elad Liebman, Maytal Saar-Tsechansky, and Peter Stone. Dj-mc: A reinforcement-learning agent for music playlist recommendation, 2014.

[27] R. Likert. A technique for the measurement of attitudes. *Archives of Psychology*, 22, 140, 1932.

[28] Patrick T. McMullen. Influence of number of different pitches and melodic redundancy on preference responses. *Journal of Research in Music Education*, 22(3), 1974.

[29] Cori L. Pelletier. The Effect of Music on Decreasing Arousal Due to Stress: A Meta-Analysis. *Journal of Music Therapy*, 41(3), 2004.

[30] Sheri L. Robb, Ray J. Nichols, Randi L. Rutan, Bonnie L. Bishop, and Jayne C. Parker. The Effects of Music Assisted Relaxation on Preoperative Anxiety1. *Journal of Music Therapy*, 32(1), 1995.

[31] Li-Yen Shue and Reza Zamani. An admissible heuristic search algorithm. In *Proceedings of the 7th International Symposium on Methodologies for Intelligent Systems*, ISMIS '93, Berlin, Heidelberg, 1993. Springer-Verlag.

[32] Gorsuch R. L. Lushene R. Vagg P. R. Jacobs G. A. Spielberger, C. D. *Manual for the State-Trait Anxiety Inventory*. Palo Alto, CA: Consulting Psychologists Press, 1983.

[33] Valerie N. Stratton and Annette H. Zalanowski. The Relationship Between Music, Degree of Liking, and Self-Reported Relaxation. *Journal of Music Therapy*, 21(4), 1984.

[34] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA, 2018.

[35] Myriam V. Thoma, Roberto La Marca, Rebecca Brönnimann, Linda Finkel, Ulrike Ehlert, and Urs M. Nater. The effect of music on the human stress response. *PLOS ONE*, 8(8), 2013.

[36] George Tzanetakis and Perry Cook. Marsyas: A framework for audio analysis. *Org. Sound*, 4(3), 1999.

[37] Catherine L. Walters. The Psychological and Physiological Effects of Vibrotactile Stimulation, Via a Somatron, on Patients Awaiting Scheduled Gynecological Surgery. *Journal of Music Therapy*, 33(4), 1996.

[38] Darcy DeLoach Walworth. The Effect of Preferred Music Genre Selection Versus Preferred Song Selection on Experimentally Induced Anxiety Levels. *Journal of Music Therapy*, 40(1), 2003.

[39] Shu-Ming Wang, Lina Kulkarni, Jackqulin Dolev, and Zeev N Kain. Music and preoperative anxiety: a randomized, controlled study. *Anesthesia and analgesia*, 94(6), 2002.

[40] Xinxi Wang, Yi Wang, David Hsu, and Ye Wang. Exploration in interactive personalized music recommendation: A reinforcement learning approach. *ACM Trans. Multimedia Comput. Commun. Appl.*, 11(1), 2014.

[41] George Kingsley Zipf. *Human behavior and the principle of least effort; an introduction to human ecology*. Addison-Wesley Press Cambridge, Mass, 1949.

# Appendix A

# Approval of the Ethics Application



**Notification of Approval**

| | |
|---|---|
| Date: | April 27, 2020 |
| Study ID: | Pro00097850 |
| Principal Investigator: | Yourui Guo |
| Study Supervisor: | Abram Hindle |
| Study Title: | Sound Relaxation: Soundscape Exploration using Reinforcement Learning |
| Approval Expiry Date: | Monday, April 26, 2021 |
| Sponsor/Funding Agency: | KIAS |

RSO-Managed Funding:

| Project ID | Project Title | Speed Code | Other Information |
|---|---|---|---|
| RES0035224 | Deep Learning for Sound Recognition | | |

Thank you for submitting the above study to the Research Ethics Board 2. Your application has been reviewed and approved on behalf of the committee.

Any proposed changes to the study must be submitted to the REB for approval prior to implementation. A renewal report must be submitted next year prior to the expiry of this approval if your study still requires ethics approval. If you do not renew on or before the renewal expiry date, you will have to re-submit an ethics application.

***Approval by the Research Ethics Board does not encompass authorization to recruit and/or interact with human participants at this time. Researchers still require operational approval (eg AHS) and must meet the requirements imposed by the public health emergency (link to Alberta COVID page).***

Sincerely,

Dr. Ubaka Ogbogu, LLB, BL, LLM, SJD
Chair, Research Ethics Board 2

*Note: This correspondence includes an electronic signature (validation and approval via an online system).*

# Appendix B

# Consent Form

Pro00097850

**INFORMATION LETTER and CONSENT FORM**

**Study Title: Sound Relaxation: "Sweet Spot" Exploration in Soundscape using Reinforcement Learning**


**Research Investigator:**
Yourui Guo
259B Computing Science Center
University of Alberta
Edmonton, AB, T6G 2S4
yourui@ualberta.ca
1.780.729.3565

**Supervisor:**
Professor Abram Hindle
4-47 Athabasca Hall
University of Alberta
Edmonton, AB, T6G 2E8
abram.hindle@ualberta.ca
1.780.492.3927


*Background*

- You are invited to participate in a research project about developing sound therapy. We will ask you to attend a session of sound therapy conducted by our system, along with attending a survey and an optional interview. Your friends or colleagues might have recommended you for this study.
- The results of this study will be used for my thesis and publication. This work is funded by KIAS.
- Before you make a decision, one of the researchers will go over this form with you. You are encouraged to ask questions if you feel anything needs to be made clearer. You will be given a copy of this form for your records.

*Purpose*

- This research aims at increasing individual's relaxation level by automatically playing sounds fine-tuned by our system. We seek to validate if the system is useful for individual's relaxation level, and if the technique can determine individual's preference on soundscapes.

*Study Procedures*

- After you agree to this study we will ask you to listen to soundscapes generated by the system. You will give feedback by pressing the “next” button to indicate that you want to switch to the next sound. One can also participate in optional interview. Total time commitment should be more than 30 minutes and less than 40 minutes.

*Benefits*

- You might benefit from this study such as learning about the algorithms and contributing to science.
- This study can benefit individuals who need sound therapy because the research can determine a personalized setting of soundscape automatically for individuals.
- Beyond your time, there is no cost to you, or compensation.

*Risk*

Pro00097850

**INFORMATION LETTER and CONSENT FORM**

**Study Title: Sound Relaxation: "Sweet Spot" Exploration in Soundscape using Reinforcement Learning**


**Research Investigator:**
Yourui Guo
259B Computing Science Center
University of Alberta
Edmonton, AB, T6G 2S4
yourui@ualberta.ca
1.780.729.3565

**Supervisor:**
Professor Abram Hindle
4-47 Athabasca Hall
University of Alberta
Edmonton, AB, T6G 2E8
abram.hindle@ualberta.ca
1.780.492.3927


*Background*

- You are invited to participate in a research project about developing sound therapy. We will ask you to attend a session of sound therapy conducted by our system, along with attending a survey and an optional interview. Your friends or colleagues might have recommended you for this study.
- The results of this study will be used for my thesis and publication. This work is funded by KIAS.
- Before you make a decision, one of the researchers will go over this form with you. You are encouraged to ask questions if you feel anything needs to be made clearer. You will be given a copy of this form for your records.

*Purpose*

- This research aims at increasing individual's relaxation level by automatically playing sounds fine-tuned by our system. We seek to validate if the system is useful for individual's relaxation level, and if the technique can determine individual's preference on soundscapes.

*Study Procedures*

- After you agree to this study we will ask you to listen to soundscapes generated by the system. You will give feedback by pressing the “next” button to indicate that you want to switch to the next sound. One can also participate in optional interview. Total time commitment should be more than 30 minutes and less than 40 minutes.

*Benefits*

- You might benefit from this study such as learning about the algorithms and contributing to science.
- This study can benefit individuals who need sound therapy because the research can determine a personalized setting of soundscape automatically for individuals.
- Beyond your time, there is no cost to you, or compensation.

*Risk*

# Appendix C

# Recruitment Letter

Hi,

My name is Yourui Guo and I am a graduate student from the Computing Science at the University of Alberta. I am writing to invite you to participate in my research study about sound relaxation exploration using Reinforcement Learning.

If you decide to participate in this study, you will be asked to listen to soundscapes selected by our system. You will need to control a remote desktop using AnyDesk and press a “next” button to indicate how much you like the current-playing soundscape. The system will change the soundscapes based on your feedbacks. Furthermore, we will ask you about the experience of the sound therapy session. The experiment session is around 20 minutes, and you will need few more minutes to fill out the survey.

This is completely voluntary. You can choose to be in the study or not. If you'd like to participate or have any questions about the study, please email or contact me at *yourui@ualberta.ca*.

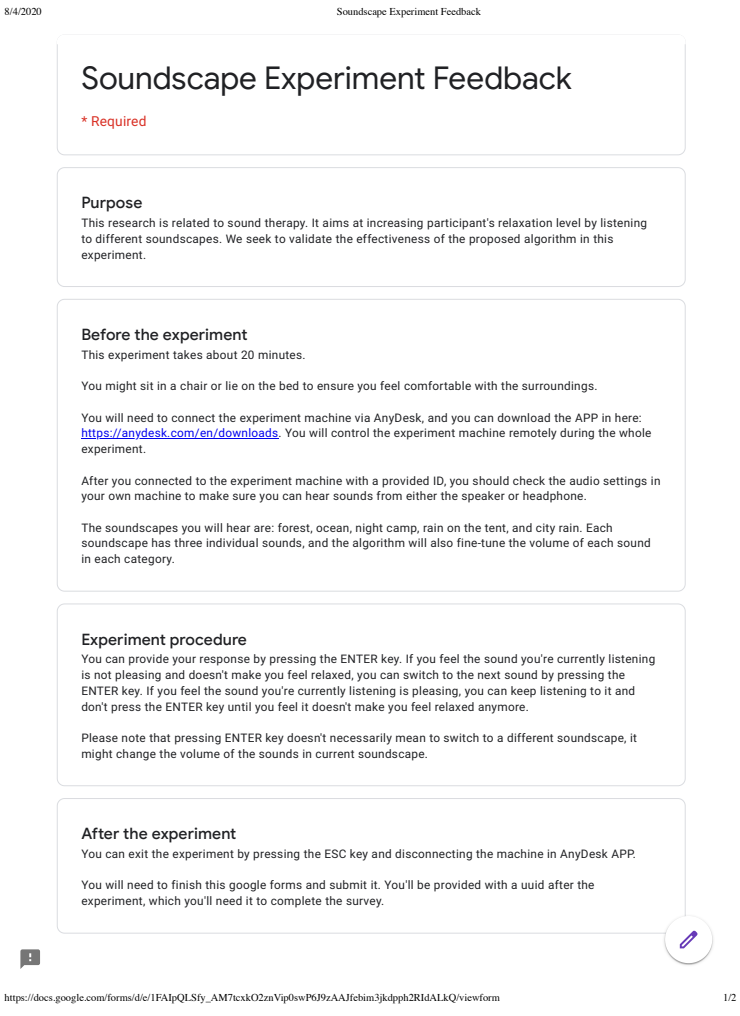
Thank you very much.

Sincerely,

Yourui Guo

# Appendix D

# Experiment Instructions



8/4/2020 Soundscape Experiment Feedback

## Soundscape Experiment Feedback

* Required

### Purpose

This research is related to sound therapy. It aims at increasing participant's relaxation level by listening to different soundscapes. We seek to validate the effectiveness of the proposed algorithm in this experiment.

### Before the experiment

This experiment takes about 20 minutes.

You might sit in a chair or lie on the bed to ensure you feel comfortable with the surroundings.

You will need to connect the experiment machine via AnyDesk, and you can download the APP in here: https://anydesk.com/en/downloads. You will control the experiment machine remotely during the whole experiment.

After you connected to the experiment machine with a provided ID, you should check the audio settings in your own machine to make sure you can hear sounds from either the speaker or headphone.

The soundscapes you will hear are: forest, ocean, night camp, rain on the tent, and city rain. Each soundscape has three individual sounds, and the algorithm will also fine-tune the volume of each sound in each category.

### Experiment procedure

You can provide your response by pressing the ENTER key. If you feel the sound you're currently listening is not pleasing and doesn't make you feel relaxed, you can switch to the next sound by pressing the ENTER key. If you feel the sound you're currently listening is pleasing, you can keep listening to it and don't press the ENTER key until you feel it doesn't make you feel relaxed anymore.

Please note that pressing ENTER key doesn't necessarily mean to switch to a different soundscape, it might change the volume of the sounds in current soundscape.

### After the experiment

You can exit the experiment by pressing the ESC key and disconnecting the machine in AnyDesk APP.

You will need to finish this google forms and submit it. You'll be provided with a uuid after the experiment, which you'll need it to complete the survey.

https://docs.google.com/forms/d/e/1FAIpQLSfy_AM7tcxkO2znVip0swP6J9zAAJfebim3jkdpph2RIdALkQ/viewform 1/2

# Soundscape Experiment Feedback

* Required

## Purpose

This research is related to sound therapy. It aims at increasing participant's relaxation level by listening to different soundscapes. We seek to validate the effectiveness of the proposed algorithm in this experiment.

## Before the experiment

This experiment takes about 20 minutes.

You might sit in a chair or lie on the bed to ensure you feel comfortable with the surroundings.

You will need to connect the experiment machine via AnyDesk, and you can download the APP in here: https://anydesk.com/en/downloads. You will control the experiment machine remotely during the whole experiment.

After you connected to the experiment machine with a provided ID, you should check the audio settings in your own machine to make sure you can hear sounds from either the speaker or headphone.

The soundscapes you will hear are: forest, ocean, night camp, rain on the tent, and city rain. Each soundscape has three individual sounds, and the algorithm will also fine-tune the volume of each sound in each category.

## Experiment procedure

You can provide your response by pressing the ENTER key. If you feel the sound you're currently listening is not pleasing and doesn't make you feel relaxed, you can switch to the next sound by pressing the ENTER key. If you feel the sound you're currently listening is pleasing, you can keep listening to it and don't press the ENTER key until you feel it doesn't make you feel relaxed anymore.

Please note that pressing ENTER key doesn't necessarily mean to switch to a different soundscape, it might change the volume of the sounds in current soundscape.

## After the experiment

You can exit the experiment by pressing the ESC key and disconnecting the machine in AnyDesk APP.

You will need to finish this google forms and submit it. You'll be provided with a uuid after the experiment, which you'll need it to complete the survey.

# Appendix E

# Survey



## Soundscape Experiment Feedback

* Required

Soundscape Experiment Feedback

Please provide the first 5 digits of uuid displayed in the terminal *

Your answer

Which soundscape did you like the most *

- 1: Forest
- 2: Ocean
- 3: Night-time camping
- 4: rain on the tent
- 5: City rain
- Other:

Do you feel the algorithm eventually chose the soundscape that makes you feel most relaxed? *

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| not really | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | yes!! |

# Soundscape Experiment Feedback

* Required

Soundscape Experiment Feedback

Please provide the first 5 digits of uuid displayed in the terminal *

Your answer

Which soundscape did you like the most *

- 1: Forest
- 2: Ocean
- 3: Night-time camping
- 4: rain on the tent
- 5: City rain
- Other:

Do you feel the algorithm eventually chose the soundscape that makes you feel most relaxed? *

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| not really | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | yes!! |