Understanding and Improving Interactive Systems Design with Human-in-the-Loop Machine Learning

ZHANG YAQIAN

School of Computer Science and Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2020

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

07-Aug-2019

Date

Zhang Yaqian

ZHANG YAQIAN

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

07-Aug-2019

.....

Date

.....

GOH WOOI BOON

Authorship Attribution Statement

This thesis contains material from [3] paper(s) published in the following peer-reviewed journal(s) / from papers accepted at conferences in which I am listed as an author.

Chapter 2 is published as Yaqian Zhang and Wooi-Boon Goh. The influence of peer accountability on attention during gameplay. *Computers in Human Behavior*, 84:18-28, 2018.

The contributions of the co-authors are as follows:

- I developed the tablet game, conducted the research trial, analyzed the data and prepared the manuscript draft.
- Professor Goh proposed the research direction, provided guidance on the game design and experiment design, and revised the manuscript.

Chapter 4 is published as Yaqian Zhang, Jacek Mańdziuk, Chai Hiok Quek, Wooi-Boon Goh. Curvature-based method for determining the number of clusters. *Information Sciences*, 415:414-428, 2017.

The contributions of the co-authors are as follows:

- I designed and implemented the algorithm and prepared the manuscript drafts.
- Professor Mańdziuk provided guidance on the experiment design, the interpretation of results, and revised the manuscript.
- Professor Quek proposed the research direction and provided guidance on the choice of publication venue.
- Professor Goh revised the manuscript.

Chapter 5 is published as Yaqian Zhang and Wooi-Boon Goh. Bootstrapped policy gradient for difficulty adaptation in intelligent tutoring systems. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 711-719. International Foundation for Autonomous Agents and Multiagent Systems, 2019.

The contributions of the co-authors are as follows:

- I developed the algorithm, conducted the theoretical analysis, and prepared the manuscript draft.

- Professor Goh proposed the research direction and research idea, and revised the manuscript.

07-Aug-2019

Zhang Yaqian

Date

ZHANG YAQIAN

Acknowledgements

First, my deepest gratitude goes to my supervisor Associate Professor GOH Wooi Boon for offering me the opportunity to pursue a Ph.D. degree in NTU and for providing continuous support throughout my Ph.D. research. Without his encouragement and guidance, this thesis would be impossible. I'm tremendously grateful for his patience and inspiration over the past four years.

I wish to thank Associate Professor CHAM Tat Jen from NTU and Associate Professor POON Kin Loong Kenneth from National Institute of Education (NIE) of Singapore for serving on my Thesis Advisory Committee (TAC) and for the insightful feedback and guidance on my research.

I would also like to thank my coauthors Professor Jacek MAŃDZIUK from Warsaw University of Technology and Associate Professor QUEK Chai Hiok from NTU for the valuable discussions and guidance on the work of Curvature-based clustering.

Thanks are also due to all the subjects in the two research trials including the forty NTU students who played Stroop game and the hundreds of anonymous participants who played the visual memory game online.

Finally, I wish to thank my husband, TU Enmei, and my parents ZHANG Shiqing and ZHANG Xueqing, for their everlasting and unconditional love.

Abstract

New developments in machine learning techniques have created opportunities for the Human-Computer Interaction (HCI) community to incorporate more intelligent means to improve and enhance user experience during the interaction. This thesis starts by exploring and identifying a suitable role where machine learning algorithms can play to improve the design of interactive systems. Once the area has been identified, a suitable learning algorithm has been designed and evaluated to ensure it is able to address the constraints posed by human-in-the-loop interactive systems.

Firstly, a user study was conducted to explore design factors that might influence users performance during competitive and cooperative gameplay. A key observation was that there is a significant performance decline when the disparity in abilities between the gaming partners is large. This result suggests that to maintain a high level of cognitive engagement, performance disparity among group members needs to be moderated. In order to reduce this disparity, stronger users should be challenged with harder tasks and less competent users should be presented with easier tasks. In short, automatic task difficulty adaption was seen to be an important area in improving the design of interactive systems and maintaining user performance. This motivates the subsequent research on how the difficulty level of a series of tasks can be autonomously adjusted based on each individual's ability, especially in relation to the design of responsive intelligent tutoring systems.

Traditionally, difficulty level is often determined by domain experts based on some hand-crafted rules. However, with the adoption of Massive Open Online Courses (MOOCs), it has become harder to manually personalize task difficulty as the system designers are faced with a very large question bank and a user base consisting of individuals with diverse backgrounds and ability levels. This research focuses on developing a data-driven method to adaptively adjust difficulty levels in order to maintain a target user performance in a visual memory task the difficulty level of which is highly variable among different individuals. The first challenge is to

obtain personalized difficulty ranking. This was addressed using a clustering-based method which can learn a personalized difficulty ranking based on pre-collected data. A novel general method for determining the number of clusters was proposed by exploiting the curvature information in the clustering objective function. Unlike existing methods that often require substantiated computational resources and parametric assumptions, the proposed approach is computationally efficient and suitable for use in real-time interactive applications.

The next challenge is the issue of difficulty adjustment which was formulated as a reinforcement learning problem. Reinforcement learning (RL) is a class of machine learning algorithms which is concerned with sequential decision making. Unlike traditional RL problems, like controlling robots (MuJoCo) or playing board games (game of Go), where accurate simulators exist, the environment being considered here is a typical HCI system that involves a human in the loop. The cost of taking a sample is thus a critical consideration as it directly affects user interaction experience and impacts the perceived responsiveness and intelligence of the interactive system. In addition, unlike many recent RL studies, the action space considered in this work is significantly larger, comprising of hundreds of different visual memory tasks. To address these constraints, a novel bootstrapped policy gradient (BPG) framework was developed, which can incorporate prior knowledge of difficulty ranking into policy gradient to enhance sample efficiency. BPG was applied to solve the difficulty adaptation problem in the challenging RL environment comprising of large action spaces and short horizon, and was demonstrated to be able to achieve fast and unbiased convergence both in theory and in practice.

Contents

List of Figures

List of Tables

Symbols and Acronyms

Symbols

p	significance level
M	mean
SD	standard derivation
$\mathbb{R}^n$	the n-dimensional Euclidean space
$\propto$	proportional relation
$O(\cdot)$	order of magnitude or ergodic convergence rate (running average)
κ	curvature
$J(\cdot)$	objective function
k	cluster number
$\mathcal{P}$	a set of users
p_i	a user
$\mathcal{A}$	a set of questions or actions
a_i	a question or an action
$DR(p_i)$	the difficualy ranking profile of user p_i
$\pi_\theta(a)$	policy with parameter θ
$\nabla_\theta J(\theta)$	the gradient with respect to θ
$\tilde{\nabla}_\theta J(\theta)$	a surrogate gradient
r_{a_i}	the reward for an action a_i
g_{a_i}	the grade for a question a_i
G	the target grade
$\mathcal{X}_{a_i}^{+}$	the better action set for an action a_i
$\mathcal{X}_{a_i}^{-}$	the worse action set for an action a_i
$\widehat{\pi}_\theta^{+}(a_i)$	the probablity of the better action set for an action a_i
$\widehat{\pi}_\theta^{-}(a_i)$	the probablity of the worse action set for an action a_i
$\mathcal{X}_a^{+'}$	the inverse better action set for an action a_i

$\mathcal{X}_a^{-'}$	the inverse worse action set for an action a_i
$Q(a)$	the value function in reinforcement learning

Acronyms

ML	Machine Learning
RL	Reinforcement Learning
HCI	Human Computer Interaction
ANOVA	Analysis of variance
RQ	Research Question
WM	Working Memory
DDA	Dynamic Difficulty Adaptation
DRP	Difficulty Ranking Personalization
UCI	University of California Irvine Machine Learning Repository
AWS	Amazon Web Service
NDPM	Normalized Distance based Performance
MAB	Multi-Armed Bandit
MDP	Markov Decision Process
POMDP	Partially Observable Markov Decision Process
PG	Policy gradient
BPG	Bootstrapped Policy Gradient
DPG	Deterministic Policy Gradient
i.e.	id est
w.r.t	with respect to

Chapter 1

Introduction

1.1 Background

1.1.1 Personalized Interactive Systems Design

One of the key goals in Human-Computer Interaction (HCI) research is to improve interactive systems design to facilitate high-quality interactive experiences. The traditional approaches evaluate several hand-crafted designs based on the experience and intuition of the designer [4, 5]. Specifically, the designers need to first put together a few design options based on their domain knowledge and then conduct some experiments like A/B testing to select a design choice that leads to the best result. The effectiveness of such design decisions, therefore, relies heavily on the domain expertise of the designers. To alleviate this issue, some research works [6–8] applied machine learning techniques, like Bayesian optimization, to automate this process by exploring the design space more efficiently. However, these approaches usually optimize a certain objective, e.g. the user engagement sampled across the whole population. After the offline analysis, a universal and static design decision is employed for all subsequent users. Such design decisions, though generally optimal, may not accommodate certain groups of users well. In fact, with the widespread use of the Internet, interactive systems are faced with the challenge of a far more diverse user base with different background and ability levels, bringing an emerging need to personalize interactive systems design to address varying individual traits (see Figure 1.1).

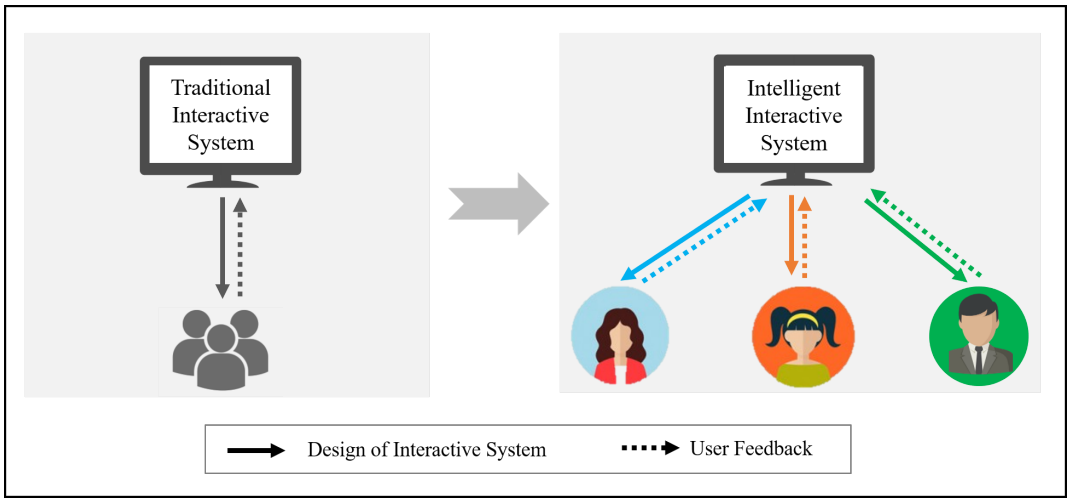


FIGURE 1.1: Static versus personalized interactive systems design.

Tailoring the behavior of the system to a particular user has the potential to make the interaction more effective and engaging. For example, in online educational systems, to make learning activities more effective, personalized education contents and pedagogical strategies can be provided to different students based on their background and learning styles [5, 9–12]. In recommendation systems, to enhance the click-through rate, different items can be fed to different users according to their individual interests [13, 14]. Similarly, in gaming systems, to enhance immersion or play retention, different displays or challenge levels can be presented to the players based on their current expertise levels [15–19].

To support personalized interactive experiences, suitable adaptation needs to be made on the design of the interactive systems to accommodate user differences. Specifically, in term of the objectives of adaption, it can be summarized from two aspects. The first is to address the individual differences among the heterogeneous user set by adapting the environment to the specific current user whom the system is currently facing. The second aspect of adaptation is to accommodate the changes within the same user as he or she develops skills and changes strategies after extended engagement. As to what is to be adapted in the interactive system design, this will depend on each application scenario. The adaptation can be applied in various design spaces, such as altering the horizontal spacing and vertical gap between pipes in the flappy bird game [7], changing the position of spawns of enemies in the shooting game [17]. Besides these numerical or continuous design spaces, adaptation can also deal with categorical design choices, like choosing command set for a gesture-based interface [20], selecting intervention messages in

an online crowd-sourcing platform [21], determining pedagogical strategies in an education system [22].

Of particular interest to this thesis is the adaptation of the challenge levels. Designers have always endeavored to present appropriate difficulty levels to the users. This topic has been discussed in various contexts, including e-learning [5, 9, 23], computerized adaptive testing [24], dynamic game balancing [18], procedure content generation [25, 26]. Based on the flow theory [27], a task being too hard may lead to frustration and a task being too easy may lead to boredom. Likewise, theoretical supports have also been provided for difficulty adaptation on the education front. The theory of the zone of proximal development (ZPD) [28] argues that providing the students with the tasks that are just beyond their current ability can scaffold the learning process. Many researchers have conducted empirical studies for difficulty adaptation. For example, in an educational game, the difficulty level of questions are dynamically tuned according to the student's grade [23]. In a fighting game, the opponent's behavior, in terms of the kicks and punches, is changed to match the player's ability level [18, 29]. In a shooting game, difficulty level is adjusted by altering the type of enemies and weapons [16]. In a racing game, the speed of the last place player is boosted up to balance the game difficulty [25].

However, when it comes to the key question of how the adaption is actually conducted, many existing implementations follow simple heuristic rules, like reducing the number of enemies by certain amount if the damage made is above some level in a shooting game [16], or increasing/decreasing the difficulty level by one for three consecutive correct/incorrect answers in an education systems [9]. These rules can be problematic in noisy and imprecise environments due to its deterministic feature. Furthermore, even though some solutions employed probabilistic models [18, 23] to control the adaptive behavior, it is unclear how to ensure the stochastic methods converge to the optimal solutions. More importantly, when it comes to describing the users' individual differences, there is a lack of considerations regarding users' strengths and weaknesses. A certain type of tasks may be particularly hard for some but easy for others. But many existing works employ a universal task difficulty ranking for all the users. For example, in [18], to determine which fighting action by the opponent are more effective, a universal fixed ranking denoted as Q value, is learned. This information is then used to help alter the opponent behavior in the fighting game to match the players' abilities. With this

approach, the relative effectiveness of a fighting action is assumed to be same for all the players, despite the fact that one player may find the kicking action harder to handle and another may consider the punching action more difficult instead. In short, the adaptation mechanisms in the previous works often use heuristic rules to guide difficulty adaptation and fail to take these personal traits into account.

This thesis aims to design a dynamic difficulty adaptation mechanism to deal with these challenges. Specifically, the proposed algorithm seeks to address four goals. First, the method needs to detect the individual difference in its richness. Instead of a scalar value, the user attributes should be depicted by a multi-dimensional representation to capture users' strengths and weaknesses. Second, rather than using static adaptation rules for all the users, the adaptation needs to have the capacity to make use of the prior information of user characteristic to personalize adaption decisions for different users. The third requirement is regarding the responsiveness or adaptation speed. Since the detection directly affects the quality of the adaptation, given a new user, the detection of the user profiles should occur quickly without extensive calibration. Also, to ensure a desirable user experience, the adaptation should achieve fast convergence within a short duration of exposure. In other words, unlike offline methods, a key requirement for online adaptation methods is to be sample efficient and to make predictions and decisions based on a limited number of gameplay observations. The last requirement is regarding the robustness of the algorithm, as there is always a high level of uncertainty associated with human behaviors. The detection method needs to overcome the possible inaccurate information within the training data and maintain good generalizability. The adaption needs to achieve stable and unbiased convergence in an environment with stochastic noise.

1.1.2 Human-in-the-Loop Machine Learning

This thesis focuses on how machine learning (ML) can be incorporated into the design of intelligent interactive systems while taking into considerations the unique challenges and constraints introduced by having the human in the loop.

Supervised learning and unsupervised learning are able to extract useful information from data and making reasonable predictions for unseen cases. This ability can be potentially used to identify and understand the individual differences among

the users in the interactive systems. In the recent decade, groundbreaking results have been achieved by machine learning in the tasks of object recognition [30] and speech recognition [31]. However, to obtain good performance, a large neural network containing lots of parameters is often used as the function approximator. To determine the values of the parameters usually relies on a great number of training data. For example, the ImageNet [30], which is widely used for training in the object recognition tasks, includes millions of images. Unfortunately, in realistic interactive applications with humans in the loop, it is challenging to build such large training datasets since there is a high cost associated with data collection. Additionally, the data collection process in the training stage may lead to poor interactive experience for the users since user adaptation is not yet in place.

Reinforcement learning (RL) is a class of machine learning algorithms that can be used to make a sequence of decisions in complex environments. This sequential decision-making ability can potentially be applied to help make design decisions in the interactive systems. RL has achieved significant success in the high-dimension continuous control of robotic locomotion [32, 33] and achieving human-level gameplay [34–36]. However, a large batch size is often needed to achieve stable convergence. For example, while applying policy gradient in reinforcement learning for simulated robotic locomotion tasks (Mujoco), the common choice of batch size is often above 1000 [32, 33, 36, 37]. Such a large batch size cannot be used in a responsive interactive application as it will result in slow adaptation to the user. Instead of batch update, incremental update, which updates the parameter immediately after receiving feedback from the user, is more practical in ensuring a desirable user experience.

Therefore, it is not straightforward to simply translate the success of existing data-hungry machine learning algorithms into real-world interactive applications. The main objective of this thesis is to develop efficient ML algorithms particularly suited for use in interactive systems design. This effort has been coined as *human-in-the-loop machine learning*. To avoid confusion in terminology, it should be noted that this phrase has been used in a different context in interactive machine learning [38–40] and interactive reinforcement learning [41] to address the idea of exploiting the human knowledge, e.g. learning from a human demonstrator to accelerate the training of ML algorithms [42], or allowing the system designer to coach and correct classifier in the design of perceptual user interfaces [38]. The main goal

of these works is to improve the learning efficiency of machine learning algorithms with the help of human expertise, whereas the research here seeks to improve the human-machine interaction with the help of machine learning algorithms. Thus, when applying RL to these two kinds of research, the reward scheme is different. Nevertheless, these two kinds of research both require effectively extract useful information from limited human feedback.

1.2 Major Contributions

This thesis makes contributions in the inter-disciplinary areas of human-computer interaction and machine learning. During the course of this research, two computerized games were designed and deployed to capture the quantitative data that measured a user's attention level and visual memory ability. Analysis of these empirical data provided interesting insights into the various factors that can influence a user's selective attention and visual memory characteristics. On the machine learning front, two sample efficient machine learning algorithms have been proposed. The first addresses issues in cluster analysis and the second addresses limitations in the existing policy gradient algorithms. These new algorithms have been specifically designed to handle many of the unique constraints in human-in-the-loop machine learning but they are also useful for general ML applications not related to interactive system design.

At the onset of this research, challenged to investigate how ML can be used to improve interactive systems design, a user study was conducted to develop deeper insights into factors that influence human performance during various human-computer interaction scenarios. Based on the findings of the user study, it was decided that dynamic difficulty adaptation (DDA) was important and useful area to address in order to support a more human-centered approach to interactive system design. Next, an online visual memory game was designed as the data gathering and algorithm evaluation platform to conduct the DDA study. Finally, machine learning-based difficulty adaptation algorithms were proposed, which includes two components as shown in Figure 1.2: one part detects and understands distinctive user characteristic and other part uses this information to adjust the difficulty levels for each individual user in a stable and responsive manner. The main contributions are summarized as follows:

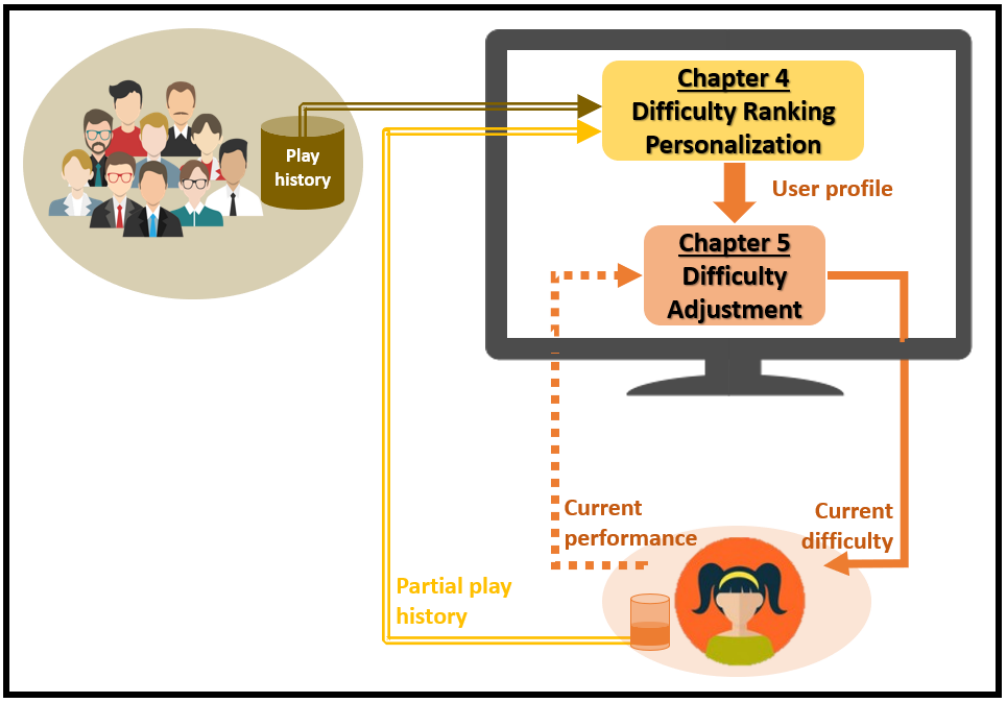


FIGURE 1.2: The overview structure of the proposed dynamic adaptation system.

- Identified the positive influence of peer accountability in enhancing attention levels and the negative effects of "large performance disparity" that provided the motivation for the subsequent DDA research.
- Designed a visual memory game as a testbed for the study of difficulty adaptation, in which a quantitative measure of the task difficulty can be obtained via in-game performance;
- Proposed a curvature-based approach for the determination of cluster number which is computationally efficient and can be used with a wide range of datasets; and introduced a clustering-based method for difficulty ranking personalization in an online visual memory game platform; and
- Proposed a framework (BPG) for incorporating prior information into policy gradient to boost sample efficiency; provided theoretical guarantee of unbiased convergence; applied BPG for stochastic difficulty adjustment in an online visual memory game platform.

1.3 Outline of The Thesis

The remaining part of this thesis is organized as follows. Chapter 2 presents a user study to investigate the influence of several design elements on users' attention. The findings of this initial study provided the motivation for the subsequent investigation into dynamic difficulty adaptation (DDA). Chapter 3 presents the rationale and design of a visual memory game which was employed as the research platform for collecting data and validating the effectiveness of the DDA algorithms described in this thesis. Chapter 4 and Chapter 5 addresses two main challenges in DDA respectively, namely how to identify the individual user differences in difficulty ranking and how to make informed and personalized difficulty adjustment. Lastly, Chapter 6 concludes this research.

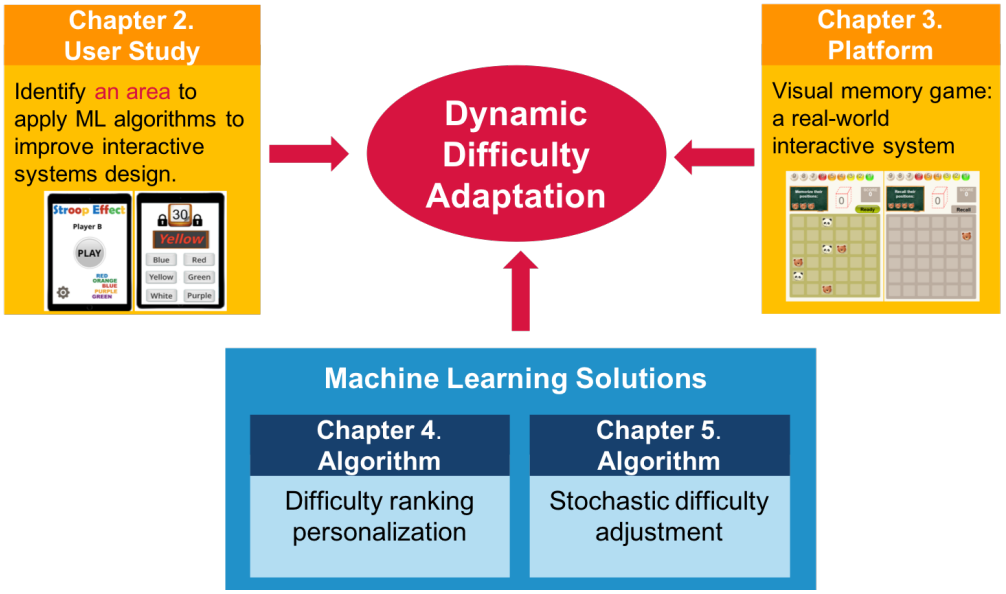


FIGURE 1.3: The outline of the main body of the thesis (Chapter 2-5).

Chapter 2

Towards Understanding Gameplay Design

This chapter [1] investigates how the design elements in the interactive systems may influence users' cognitive behaviors during gameplay. In particular, the user's selective attention, which is a key component that determines their level of engagement during the interaction, is the main focus here. A user study based on Stroop test [44, 45] was conducted to explore how selective attention level is affected by peer accountability, performance disparity, and physical distance during multiplayer gameplay. Section 2.1 presents the related works in multiplayer gameplay, peer accountability and selective attention. Section 2.2 introduces the stimuli design and the research methodology used in the research trial. The research findings are discussed in Section 2.3.

2.1 Background

To increase motivation and engagement in gameplay, competitive and cooperative game mechanics are often employed in the interactive systems design. Many researchers have studied their comparative effectiveness and influence on players' performance. For instance, competition and cooperation have been studied from the viewpoint of goal structures and the cooperative goal structure was found to result in higher motivation and effort in a motor-centered activity [46]. Others examined

[1] This chapter is published as [43].

how scoring mechanisms based on principles of collaboration and competition impact the accuracy and engagement of players in commonsense knowledge collection tasks [47]. Their results show that the competition-based scoring mechanism maintained the accuracy and increased engagement. Besides goal structures and scoring mechanisms, cooperative game design patterns can take a variety of other forms in game mechanics, such as shared goals, synergies between abilities, complementary roles, and so on [48]. Comparative studies have so far yielded mixed results due to the use of different cooperative design attributes and it remains unclear what influence the various cooperative design features or confluence of features may have on motivation and engagement. In addition, performance in a cooperative setting can be often complicated by variations in the group composition such as the ability disparity between members in the team [49] and space features such as the physical proximity between partners [50].

Of particular interest in this study is the influence of peer accountability on players' behavior. Studies in organizational behavior research have shown that accountability, either as a threat or an opportunity, has a wide range of influence on cognitive activities such as emotional labor, focus, opinions, perceptions, and attentiveness [51, 52]. In a purely competitive multiplayer gameplay, the players are only accountable to themselves. In cooperative gameplay, on the other hand, the players are not only accountable to themselves but also accountable to their teammates. The model of social judgment and choice [53] argues that accountability can serve as a fundamental force to drive a person's behavior and decisions because individuals are concerned about their self-image and status in the eyes of others. The impact of accountability has been extensively addressed in psychology and organizational behavior research [54–56] but studies examining the role of accountability on multiplayer behavior in game design are still wanting, especially in the context of comparison between competitive and cooperative gameplay, and is therefore the focus of this study.

In order to design this comparative study on cooperative gameplay which emphasizes peer accountability and contrasts its influence relative to a competitive equivalent, care must be taken to minimize the influence of other cooperative design attributes. Firstly, the task in which the players' performance is measured during gameplay must remain essentially the same in both the cooperative and competitive modes. In this way, the measurements obtained in the two gameplay

modes can be compared directly and without bias. Secondly, while maintaining the constraint mentioned earlier, the notion of peer accountability must be embedded in the game mechanics and made explicit to the cooperating players. To achieve these goals in the cooperative game design, the progress of one player is made dependent on the other in a conjunctive manner but not their individual performance measure. In particular, a competitive and cooperative version of a game was designed, which employed a simple cognitive task, namely the Stroop task [44]. This task has been widely used in neuropsychological studies as a measure for selective attention as the task can be error-prone and requires sustained attention for fast and accurate performance. In addition, the straightforward Stroop task game was intentionally kept basic and uninteresting. As pointed out in [57], a task that is inherently interesting requires little effort in soliciting the player's attention. In contrast, an uninteresting task requires the player's conscious effort to motivate attention in order to stay in the game. Using such stimuli, the measured attention levels of the players are likely to be dominated by the social influences of the competitive and cooperative attributes rather than the game itself.

Attention is an important component of engagement and immersion [58]. Attention is also strongly related to learning outcomes in the educational context [59]. Thus measuring the players' performance in terms of their attention level may shed some light on a wider question of whether a competitive or cooperative scenario will be more effective in stimulating and sustaining players' engagement during gameplay. However, the scope of attention can be either broad or narrow and each leads to a different mode of engagement [57]. During an exploratory activity such as doing flower arrangement, one's attention is broad in scope and engagement is mostly driven by curiosity. In contrast, the delicate task of hammering a small nail will narrow and focus the scope of one's attention to the exclusion of other irrelevant competing stimuli. The scope of attention addressed with the Stroop task design is narrow and as such, the results presented in this study can be generalizable to activities with well-defined performance measures such as response time, accuracy, correctness and high scores.

Additionally, the Stroop task design allows us to make quantitative measures of the players' cognitive state in terms of the players' speed and accuracy in answering each question. Such measures are more objective and do not depend on players' subjective perception and experiential recall when they are asked to rate the

attention levels of different gameplay modes in post-trial questionnaires. Unlike subjective measures, individualized quantitative measures acquired during gameplay also allow us to analyze several interesting gameplay behaviors of the players. Firstly, the temporal variations in each player's response time allow us to study not only the overall attention level of the players but their ability to sustain this attention during the gameplay duration. Secondly, the individualized performance measures during gameplay permit us to investigate the differences in behavior between the stronger and weaker performers in a cooperative setting and the influence such disparities has on the players' attention levels.

In summary, this study has several related objectives. The first research interest is to compare players' engagement in terms of overall attention level and sustained attention when playing a cognitive-oriented game in a competitive play mode and in a cooperative play mode with strong peer accountability. Secondly, given that the performance disparity between cooperating partners is known to have an impact on player's behavior, this study investigates the difference in engagement levels between the stronger and weaker performers and what influence the extent of this performance disparity has on the ability of the players to sustain their attention throughout the gameplay. Finally, given that such cooperation can be performed either remotely (apart) or in a co-located (close proximity) manner, this study also set out to investigate if physical proximity has any significant influence on players' attention level during cooperative play.

2.1.1 Accountability

In the social psychological literature, individual-level accountability is defined as "*an implicit or explicit expectation that one's decisions or actions will be subject to evaluation by some salient audience(s) with the belief that there exists the potential for one to receive either rewards or sanctions based on this expected evaluation.*" [54]. Accountability to others has been regarded as an important social psychological link between individuals and social systems [60]. It should be noted that the peer accountability discussed in this study refers to felt accountability, which is focused on the actor's subjective interpretation of accountability from a phenomenological view of accountability [56] as opposed to the attribution of accountability from audiences' point of view [55].

Empirical research has shown that accountability can influence people in many ways, including cognition, behavior, affective states and decision making [51]. For instance, the studies on social interdependence theory suggested that it is crucial for educators to make sure that the individuals' outcomes are affected by each other's actions and each student is held accountable in order to promote effective cooperative learning [61]. However, the consequences of accountability are not always beneficial. High level of accountability was also found to be associated with some negative outcomes, such as "*higher depressed mood at work, lower levels of organizational commitment and work intensity, and decreased job satisfaction*", especially when there is a low level of fit between the person and organizational environment [62].

When it comes to cognitive activities in particular, studies in organizational behavior research have found that accountability can affect what people think (e.g. preferences) and how people think (e.g. reasoning) [52]. Yet there is little research in multiplayer gameplay investigating how the accountability to cooperating peers might influence a player's cognitive state. In cooperative gameplay, the existence of teammates as salient audiences can exert extra responsibility and accountability and might alter players' cognition and behaviors when the actors explicitly or implicitly regard these "accounts" as part of their self-images to protect and enhance, as pointed out by the model of social judgment and choice [53]. Therefore, one of the goals in this study is to investigate the influence of peer accountability on players' attention in a multiplayer gaming environment.

Furthermore, the complex relationship between accountability and cognition can be moderated by many other factors such as the characteristics of the audience. For example, while being required to give opinions on a controversial issue, subjects tended to shift their views towards the position that they thought the audiences held [60]. And other studies found that the cognitive effort participants spent on the discussion under accountability pressure were related to their partner' relative expertise [63]. Specifically, when the subjects thought their partners processed a similar level of expertise on the topic as themselves, they were observed to give more cognitive effort. Based on this interesting observation, another goal of this study is to investigate if a player's cognitive state during the cooperative gameplay will be affected by their partner's relative performance level.

2.1.2 Selective Attention

Research on competition and cooperation has been conducted with various gameplay genres, like motor-centered, mathematical and brainstorming games, and they involved different application domains, such as serious games, educational games, games with a purpose (GWAPs), etc. [46, 47, 64, 65]. This study is focused on a simple cognitive game with no particular application domain, where sustained selective attention is required for good performance. Selective attention refers to the ability to attend selectively to certain aspects in a situation, while simultaneously ignoring irrelevant information that is also present. We use this in our daily interactions as it is impossible to give attention to every stimulus in the environment.

In 1935, J. R. Stroop (Stroop, 1935) published his landmark work on attention and interference in which the Stroop effect was proposed and it has since been widely used as a measure for selective attention in numerous psychological studies. In the Stroop Color-Word Interference Test, the subjects will see a series of words and are required to name the color each word is printed in, instead of what the word spells. Research findings observe that when the color of the ink does not match the name of the color (incongruent condition), the subjects take a longer time and are more prone to errors in naming the color than in the congruent condition. The Stroop test involves two stimuli, one is the target (color), and the other is the distractor (word). While facing two stimuli, attention is needed to decide whether to attend to the ink color or the word analyzer when each leads to a different potential response. Generally, performance on the Stroop task is taken as the "golden standard" for selective attention [45] and has been widely used in the neuropsychological study as a measure for selective attention in studying individual differences, drug effects, and so on. For example, to investigate functional anatomy of attention, measurements to the changes in regional cerebral blood flow were taken while subjects were performing the Stroop test [66].

2.1.3 Multiplayer Gameplay: Competition versus Cooperation

Numerous studies have investigated the relative influence of competitive and cooperative designs on players' behavior. In physically-oriented exergames, competitive play has been found to increase energy expenditure and aggression, while cooperative play has been found to increase motivation, pro-social behaviors and promote continued play [67]. A study related to stress reduction observed a similar decline of stress levels in the competitive and cooperative gameplay. But the competitive condition led to a slightly less positive impression of the opponent for the participants [68]. Some positive social benefits of playful competitive gaming were also reported in a longitudinal study, such as decreasing of conduct problems and improvement of peer relations [69]. Besides these psychological and social effects of cooperative and competitive gameplay, some studies also examined their physiological influence. While playing an action game (Bomberman), subjects exhibited higher levels of physiological activities (facial EMG, respiration, electrodermal and cardiac activities) in competitive play than cooperative play [70]. When it comes to motor performance, an extensive meta-analysis of the relative impact of cooperative, competitive and individualistic efforts on motor skills tasks suggests that cooperation is the mode that promotes higher performance on motor skills tasks under most conditions [71]. Another more recent study on a motor activity-centered computer game partially supports this argument in the sense that they observed cooperative goal structure can lead to greater motivation to put efforts in the game compared to the competitive version, yet no significant differences in performance were observed [46]. In addition, some studies examined some factors that may moderate the impact of competitive and cooperative gameplay, such as performance feedback, players' pre-existing relationship, team ability disparity, etc. A study on the influence of performance feedback during casual online gameplay found that players had a more favorable perception of their partners when winning cooperatively and their competitors when losing competitively. However, they rated their cooperating partners less favorably when they lost together and their competitors when they beat them. [72]. As for pre-existing relationships, cooperating with friends was found to result in a stronger goals commitment than partnering with strangers [46]. However, relationships did not have any significant influence on the participants' feelings of hostility or cooperative behaviors after

competitive and cooperative gameplay in a violent video game [73]. As for team ability disparity [49], the performance in a competitive reward structure was higher than in a cooperative structure for teams with large disparity. However, when the disparity is small, no significant difference was observed. Furthermore, studies also reported the socio-cognitive functions of space features for co-located collaboration settings [74]. For instance, the majority of participants felt that sitting close together with partners was more effective and more enjoyable for collaboration since communication was easier [50]. Besides the convenience for initiating and conducting conversations, close proximity can also help maintain task and group awareness [74].

In the area of education, it has long been established that interaction with peers is an effective way of developing skills [28] and that knowledge construction is a social and collaborative process [75, 76]. Works on cooperative learning in the classroom context suggest peer collaboration can have a positive influence on learning outcomes as well as on general attitude of learners [77, 78]. For example, a study employing cooperative and competitive versions of the Wii games in a classroom setting found that cooperative games can benefit the social interaction of students with behavior and learning difficulties by increasing their classroom interaction frequency [77]. However, it should be noted that such positive effect of cooperative learning does not always occur by simply placing students in groups. In fact, from the motivational perspective, in order to make cooperative learning effective and successful some key components, like shared goals and individual accountability, are quite essential [79]. Other factors, such as group composition [80], and instructional material in terms of the type of knowledge involved [81], may also affect the effectiveness of cooperative learning. Besides collaboration, the advantages of competition have also been reported in some research studies. In an educational mathematics game-based study, although both competitive and cooperative modes stimulated greater situational interest and enjoyment compared to individual play, only the competitive play mode was found to increase game performance. The collaborative play mode on the other hand, had a higher re-engagement potential [64].

Other researchers have studied how cooperative and competitive elements can be employed to improve productivity. Such applications include labeling data, collecting commonsense knowledge, etc. For example, one study examined how the

outcomes of crowdsourcing are affected by social transparency and different peer-dependent reward schemes [82]. The results have shown that social transparency applied to a collaborative scheme that rewarded the collective output of the paired workers helped reduce social loafing through peer accountability. On the other hand, social transparency applied to a scheme that rewarded workers based on how much they can outperform the other actually increased the incentive to compete, thus increasing their output relative to those who work individually. These results suggest that an appropriate peer-dependent reward scheme design can motivate higher output from workers. Another study investigated the influence of competitive and cooperative visualizations on performance, pressure, balance of participation for group mirrors in a brainstorming session [65]. Their findings suggest that visualizations having a mixture of competitive and cooperative features stimulated the highest productivity and satisfaction in terms of the number of ideas generated during the brainstorming session.

As can be gleaned from the findings of the many previous works, the comparative advantages of competition and cooperation seem varied, inconsistent and dependent upon various factors such as team composition, task type and dimension, etc [83]. A majority of the past research reviewed presented analysis from data derived from subjective questionnaires. In contrast, this study has employed quantitative measures to analyze the differences in players' attention during competitive and cooperative gameplay. Such dense and sensitive measures permit us to carry out experimental analysis related to the disparity in performance abilities between each pair and players' temporal degradation in performance during gameplay.

2.1.4 Terminology and Notation

Several notations adopted in this study need clarification. In traditional game theory, games are divided into two basic types: competitive and cooperative. However, in the game design community, a third type, collaborative games has been differentiated as "*Cooperative players may have different goals and payoffs whereas collaborative players have only one goal and share the rewards or penalties of their decisions.*" [84]. Based on their definition, the cooperative design in this study is actually collaborative in nature because the two partners have one collective goal and share the rewards or penalties. However, to maintain consistency in notation

with most related works in the literature, the word cooperation is adopted in this thesis.

Besides competition and cooperation, some works combine the two to examine the effectiveness of inter-group competition. Investigations on the influence of pure competition, pure cooperation and inter-group competition on motor performance in a basketball free-throw activity found that inter-group competition led to higher levels of intrinsic motivation and performance [85]. In the pure cooperation condition, the team's goal is to exceed a target number (computed as the mean of previous individual scores) while in the inter-group competition condition, the team's goal is to exceed another team's score. Based on this definition, it should be noted that the cooperative mode in this study is effectively inter-group competition. In short, the goal of the competitive gameplay mode is for each individual to beat the other player and the goal of the cooperative gameplay mode is for the pair to work together to beat the scores of the previous team.

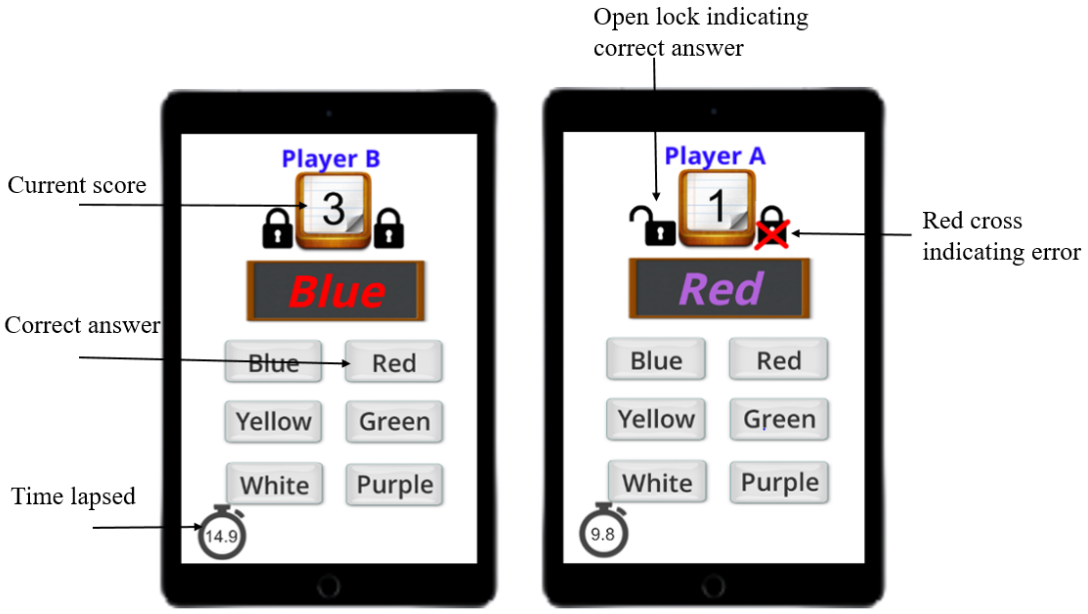
2.2 Method

2.2.1 Participants

Participants ($N = 40$) comprising of undergraduates and graduates were recruited from Nanyang Technological University. Their ages ranged from 20 to 41 years ($M = 26.41$, $SD = 5.34$). The data collection was conducted after the approval of the NTU-Institutional Review Board (IRB-2016-10-001).

2.2.2 Stimuli

Based on the Stroop effect, a tablet computer-based game was designed focusing on the incongruent condition with unmatched color and word pairs. During the game, the players see a set of words. The colors of the words do not match the words' meaning, such as shown in Figure 2.1(a), the word "Blue" is in red-colored fonts. Players are required to select the matching color of the words. For example in Figure 2.1(a), the correct answer is "Red". The goal of the game is to complete as many questions correctly as possible within the 90 seconds allocated per round.



(a) Competitive mode (b) Cooperative mode

FIGURE 2.1: Multiplayer game based on the Stroop effect: (a) The game screen of player B in the competitive mode. The locks are inactive. (b) The game screen of player A in the cooperative play, where player A has completed correctly (left lock open) but partnering player B made a mistake (right lock shows a red cross)

This multiplayer game is designed with two gameplay modes, namely cooperative and competitive modes. To ensure the two players start the game simultaneously, a third tablet computer (acting as a server) was used by the facilitator to initiate the game when both the players were ready to start.

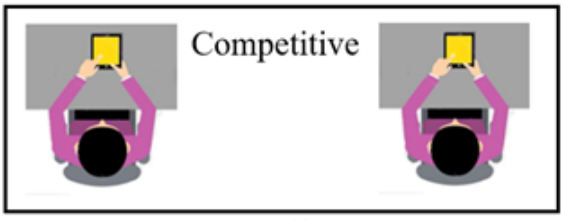
In competitive gameplay, the goal of each player is to surpass the score achieved by the other. As seen in Figure 2.1(a), the current number of correct answers is shown on the top of the screen and the elapsed gameplay time is shown on the bottom left. The two players play independently. After 90 seconds is over, the one who completed the most questions correctly will be announced as the winner with a "Win!" flashed on her tablet, while the other player receives a "Lose". Both receive a "Win!" in the event of a tie.

In cooperative play, the goal of the game is to beat the score the previous participating team achieved during their corresponding round. The team will be informed whether they have won or lost at the end of each round. Team members cooperate in the following manner. Both will receive the same question simultaneously. As seen in Figure 2.1(b), there are two locks on display. Each player controls one lock.

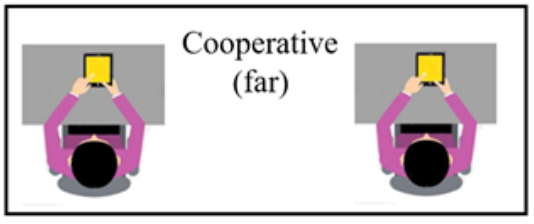
More specifically, if one finishes the question correctly, he or she will see the left lock open with the sound of a delightful chime and if his or her partner also finishes the question correctly, he or she will see the right lock open as well. However, if one team member answers the question incorrectly, a red cross will appear on the corresponding lock, as seen in Figure 2.1(b), along with the sound of an error buzzer. Only when both players make their selection, can they move on to the next question. If any of the two answers is incorrect, the team gets no points.

Note that in the design of this cooperative game mechanics, the cognitive task for each individual player remains identical to that in the competitive mode, which is essentially the Stroop test task. As such, we can make a fair comparison of the performance data collected in all the gameplay modes. However, peer accountability is intrinsically embedded in the cooperative design since one player's error will compromise the high score the team can attain. Moreover, since the next question is only given when both players complete their selection, the reaction time of the slowest player has the most influence on the high score the team can achieve in 90 seconds. Peer accountability is also made visibly explicit to the players through the display of two open locks animation and "red cross" error indicator.

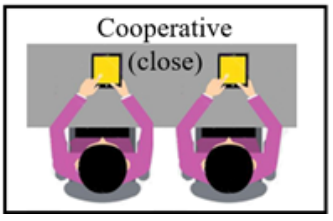
2.2.3 Procedure



(a) Competitive mode



(b) Cooperative mode - apart



(c) Cooperative mode - close

FIGURE 2.2: The three modes: (a) competitive mode, (b) cooperative mode - apart, and (c) cooperative mode - close.

Three modes were investigated in this experiment, as shown in Figure 2.2. There was the single competitive mode shown in Figure 2.2(a) and the two cooperative modes, both with the same gameplay mechanics except for the physical arrangements of the two partners. The first is cooperative mode - apart and the other is cooperative mode - close, as shown in Figures 2.2(b) and 2.2(c) respectively. The 40 participants formed 20 team pairs. Within-subject design was used with the teams playing three different modes in counter-balanced order. On arrival at the laboratory, the participants filled out a consent form and answered some profiling questions before the facilitator started the experiment. Firstly, there was an individual practice session which allowed the participants to familiarize themselves with the Stroop test task. In this practice session, the participants were required to finish ten questions correctly. After the practice session, the participants were asked if they needed more practice. If not, they started the formal experiment where the participants did three sessions using their assigned order. Each session consisted of two rounds of 90-second duration gameplay. Before the cooperative gameplay session, there was also an additional practice round to help players understand the cooperative gameplay design.

Participants also answered two different survey questionnaires, one per in-between sessions rest period and in the sequence depending on their counter-balanced order. They were asked to rate their effort, focus and preference as soon as they were able to compare the competitive-vs-cooperative modes or the apart-vs-close proximity modes. Written feedback and comments were also collected.

2.2.4 Measures

There are two common performance parameters used in studies that employ the Stroop test to measure a subject's attention level or ability. The Stroop effect is a demonstration of interference in the reaction time of performing a Stroop task. As such, a subject's reaction time during a Stroop test is widely used to gauge performance. The other is the number of erroneous selections made during the series of Stroop tests. For example, both time and error-related measures in the Stroop color-word test was successfully used to discriminate between children with fetal alcohol syndrome and healthy children [86].

In this study, the reaction time and error rate were both used to measure the attention level. Reaction time is defined as the time elapsed from the question presented to the question being answered. And error rate is defined as the ratio of the number of incorrect answers to the number all the answers completed. Players who do not pay attention to the task is expected to take a longer time to complete the task and is more likely to make mistakes than when they are focused on the task at hand.

The game score, which counts the number of correct answers completed in 90 seconds, is actually a performance measure that combines the influence of both the reaction time and error rate. Unfortunately this measure cannot be used in the analysis because in the cooperative mode, partners need to wait for each other in order to progress in the game thus providing unfair performance advantage to the individual competitive mode. However, individual reaction time in the cooperative mode is independent of the partner's performance as it does not include the waiting time. It is therefore a more accurate measure of a player's attention level in both modes of gameplay. Cooperative mode error rates are also individual performance measures as the system records the answers each player chooses independently. Their erroneous selection only affects the team's progress in the game but not the team mate's own choice.

2.3 Data Analysis and Results

2.3.1 Competitive versus Cooperative Modes

Research findings have shown that compared to individual gameplay, competitive play and cooperative play can lead to a higher level of engagement. However, results comparing competitive and cooperative gameplay have been mixed [46, 47, 64]. In this study, the following research question is proposed:

RQ1: Which multiplayer gameplay mode, competitive or cooperative, results in a higher level of attention in a cognitive task?

To answer this question, each player's performance in each gameplay mode in terms of error rate and average reaction time was computed. Then paired t-test was used to compare the results from the competitive mode and the cooperative

mode - apart in order to maintain the physical arrangement attribute constant. The differences in error rates between the competitive and cooperative modes are significant, ($t(39) = 2.673$, $p = 0.011$). Participants made relatively less errors during cooperation ($M = 2.20\%$, $SD = 3.40\%$) than during competition ($M = 3.23\%$, $SD = 4.35\%$).

Intuitively, the reduced error rates are expected to be achieved at the expense of slower reaction time. However, there was no significant difference between the reaction times in the competitive and cooperative modes, ($t(39) = 1.262$, $p = 0.215$). In fact, the average reaction time in the cooperative mode ($M = 1.049$, $SD = 0.119$) is even lower compared to that in the competitive mode ($M = 1.083$, $SD = 0.219$). In other words, the cooperative mode actually led players to make fewer errors without slowing down their response time. In performing a task like the Stroop test, this implies more cognitive effort and focus was being employed by the players during cooperation.

2.3.2 Temporal Performance Changes

The previous analysis used the average reaction times and error rates to analyze the attention level for the entire game duration. Besides the overall attention levels, the players' ability to sustain attention during gameplay was also investigated with the following research question:

RQ2: Is there any difference between the attention levels at the beginning and ending stages of the game? And if so, how does it differ in the different gameplay modes (competitive vs. cooperative)?

To answer this question, the average performance measures at the start of the game (first 20% responses) were compared with those at the end of the game (last 20% responses). A two-factor (game stage $\times$ gameplay) repeated measures ANOVA was used in this study for the reaction time. A main effect of game stage was found ($F(1, 39) = 14.881$, $p < 0.001$). No main effects of gameplay mode or any interactions were observed (all $p > 0.08$). Follow-up paired t-tests on main effect of game stage show that the reaction times increased from the start stage ($M = 1.035$, $SD = 0.186$) to the end stage ($M = 1.083$, $SD = 0.174$) in the cooperative modes ($t(39) = -3.248$, $p = 0.002$). In contrast, the change of average reaction times

in the competitive mode was not significant ($t(39) = -1.242$, $p = 0.222$), with ($M = 1.060$, $SD = 0.194$) at the start to ($M = 1.080$, $SD = 0.211$) at the end. These results suggest that the temporal performance decline occurred only in cooperative mode but not in the competitive mode.

2.3.3 Faster and Slower Players

A common feature in cooperative settings is the differences in general performance levels among the partners. Some can be faster at the task than others. Group role, that is, being the faster or the slower player in the group may be a factor that can impact the player's focus and effort, since the need to cooperate can stir up different types of feelings among players with different group roles. Faster players may feel impatient waiting for their slower partners whilst the slower counterpart may feel pressured and anxious. Therefore the following research question was considered:

RQ3: Will group role (i.e. the faster or slower players) affect attention level change in competitive and cooperative modes?

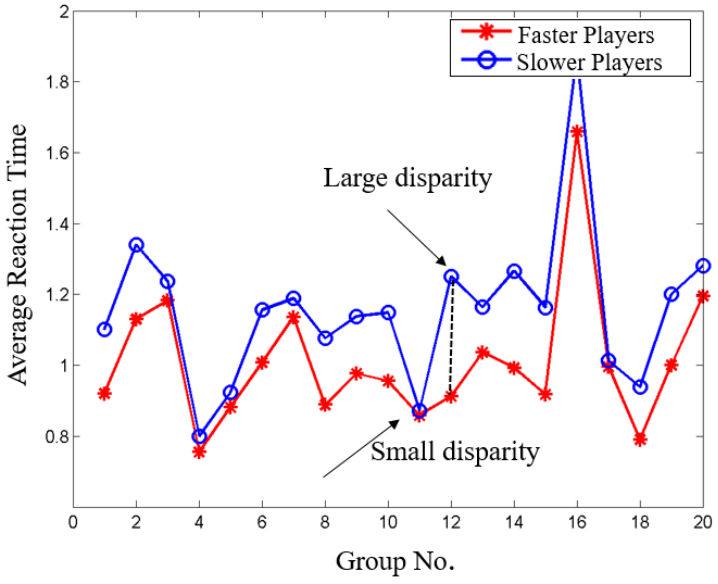


FIGURE 2.3: Identification of a player's group role using average reaction time during the competitive gameplay. Examples of pairs with large and small average reaction time disparities are group numbers 12 and 11 respectively.

Adopting the group role notion, the players in each group pair were identified as the faster or slower player based on their respective average reaction time during the competitive mode, as seen in Figure 2.3. The performances of the faster and slower players during cooperation were separately investigated. The two-factor mixed analysis of variance (ANOVA) with group role types (faster vs. slower players) as between-subject independent variable and gameplay modes (competitive vs. cooperative) as within-subject independent variable was conducted for error rate and reaction time.

Regarding error rate, there was a main effect of gameplay modes ($F(1, 38) = 7.967$, $p = 0.008$) found, but no main effect of group role ($F(1, 38) = 2.584$, $p = 0.116$) or interaction effect ($F(1, 38) = 3.204$, $p = 0.081$) were observed. Follow-up paired t-tests were performed to examine the main effect of gameplay modes for error rate in more detail. The results show that for faster players, there was no significant difference in error rate between the competitive ($M = 1.98\%, SD = 1.62\%$) and cooperative gameplay ($M = 1.61\%$, $SD = 2.17\%$) modes, ($t(19) = 0.951$, $p = 0.354$). However, the slower players made significantly fewer errors in the cooperative mode ($M = 2.78\%$, $SD = 4.28\%$) than in the competitive mode ($M = 4.48\%$, $SD = 5.74\%$), ($t(19) = 2.628$, $p = 0.016$). When it comes to reaction time, a main effect of group role ($F(1, 38) = 5.860$, $p = 0.020$) was observed as expected, but no significant main effect of gameplay modes ($F(1, 38) = 1.625$, $p = 0.210$) or interaction effects ($F(1, 38) = 1.798$, $p = 0.188$) were observed. Paired t-test also revealed that there was no significant difference in reaction time between the competitive and cooperative modes for both faster players ($M = 1.010$ to 1.011, $SD = 0.194$ to 0.119, $t(19) = -0.051$, $p = 0.960$) and slower players ($M = 1.157$ to 1.087, $SD = 0.222$ to 0.109, $t(19) = 1.712$, $p = 0.103$). These findings imply that cooperation can help the slower players improve their accuracy in the cognitive task and this improvement is not at the expense of the faster players' performance.

Another analysis was conducted to examine the temporal performance changes for the faster and slower players. Paired t-test revealed that in the cooperative mode, the temporal degradation of reaction times was significant for both the faster players ($M = 0.978$ to 1.034, $SD = 0.127$ to 0.132, $t(19) = -2.654$, $p = 0.016$), and slower players ($M = 1.042$ to 1.108, $SD = 0.113$ to 0.126, $t(19) = -3.204$, $p = 0.005$). However, in the competitive mode, there was again no significant

decline in reaction time for both the faster players ($M = 1.014$ to 1.021, $SD = 0.194$ to 0.223, $t(19) = -0.387$, $p = 0.703$) and slower players ($M = 1.106$ to 1.139, $SD = 0.187$ to 0.185, $t(19) = -1.248$, $p = 0.227$). These results suggest that the players' performance in terms of reaction time degraded over time during the cooperative gameplay. And this degradation occurred for both the faster and slower players. Fatigue could be a possible explanation if not for the fact that this temporal performance degradation should be equally applicable during the competitive gameplay but is not significantly present. Some other factors could be at play here.

2.3.4 Large and Small Performance Disparity

An interesting question to ask is what aspects of cooperative gameplay could cause the observed temporal performance decline. One possible reason is the performance disparity between the partnering pairs. If the average reaction time between the faster and slower player is large, the faster player is frequently waiting for the partner to finish and may feel bored. On the other hand, the slower player may feel nervous and pressured when the partner's lock repeatedly opens way before he is able to answer. Sustained boredom and anxiety could both impair focus and exert a negative effect on the player's attention level, leading to a declining performance with time. Based on this speculation, an additional research question was explored:

RQ4: Will performance disparity between partners affect temporal performance changes during cooperation?

The performance disparity for a group is defined as the absolute difference between the average reaction time of the two players during the competitive gameplay. This disparity between partners can vary significantly, as seen in Figure 2.3. Performance disparity in the pair in group 12 is very large whilst that in group 11 is negligible. In order to answer research question RQ4, all 20 group pairs were sorted based on increasing performance disparity, as shown in Figure 2.4. The temporal performance degradations of the groups with large performance disparities (top 40%) were compared with that of the groups with small performance disparities (bottom 40%). A 2 x 2 (performance disparity $\times$ game stage) mixed ANOVA was conducted for reaction time in cooperative mode - apart. Only a main effect

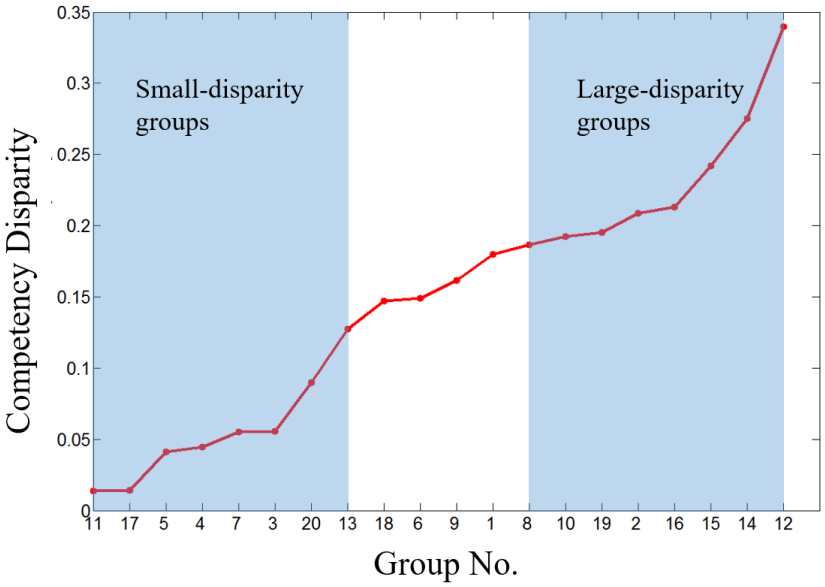


FIGURE 2.4: Groups being sorted based on increasing performance disparity.

of game stages was found ($F(1, 30) = 8.042$, $p = 0.008$). No other main effect ($F(1, 30) = 0.289$, $p = 0.595$) or interaction effect ($F(1, 30) = 0.860$, $p = 0.361$) were observed. Follow-up paired t-tests were performed to examine the main effect of game stage for large and small disparity groups in more detail. For those groups with large performance disparity, paired t-test suggested significant temporal performance decline occurred from the start stage ($M = 1.006$, $SD = 0.117$) to the end stage ($M = 1.064$, $SD = 0.136$), ($t(15) = -2.505$, $p = 0.024$). Interestingly, for those groups with small performance disparity, such decline was not significant ($M = 0.997$ to 1.032, $SD = 0.112$ to 0.092, $t(15) = -1.559$, $p = 0.140$). These results imply that the player's performance seems to degrade with time during cooperation only when there is a large mismatch in ability between partners in solving the given task in a timely fashion.

2.3.5 Close versus Apart Sitting Arrangements

Research findings on proxemics show that interpersonal physical distance can influence the interaction between people [87, 88]. In cooperative play, sitting in close proximity can create an environment that supports both verbal and non-verbal communications. As such, the potential for a stronger sense of cooperation and team morale is facilitated. There is however limited research examining the effects of physical distances during the cooperative gameplay, especially those involving

a cognitive task that requires focused attention. Therefore, the following research question was investigated:

RQ5: How does physical proximity influence players' attention levels during the cooperative gameplay?

Paired t-test was used to compare the players' performance while cooperating in the apart and close sitting arrangements seen in Figures 2.2 (b) and (c), respectively. The t-test results revealed that there was no significant difference between the sitting apart ($M = 1.049$, $SD = 0.119$) and sitting close ($M = 1.065$, $SD = 0.170$) in terms of average reaction time ($t(39) = -1.029$, $p = 0.310$). Similarly, in terms of error rate, there was also no significant difference found in sitting apart ($M = 2.20\%$, $SD = 3.40\%$) and sitting close ($M = 2.59\%$, $SD = 2.63\%$), ($t(39) = -1.080$, $p = 0.287$).

In terms of temporal performance changes, a two-factor (game stage $\times$ sitting arrangement) repeated measures ANOVA were performed for the reaction time. Main effects of game stage were found ($F(1, 39) = 27.332$, $p < 0.001$). No main effects of sitting arrangement or any interactions were observed (all $p > 0.08$). Similar to cooperative mode - apart, follow-up paired t-test shows that the reaction times also increased in the cooperative mode - close ($t(39) = -4.187$, $p < 0.001$) from the start stage ($M = 1.010$, $SD = 0.123$) to the end stage ($M = 1.071$, $SD = 0.133$). Moreover, this temporal performance degradation was again significant for both the faster players ($M = 0.999$ to 1.038, $SD = 0.184$ to 0.178, $t(19) = -2.219$, $p = 0.039$) and slower players ($M = 1.071$ to 1.127, $SD = 0.185$ to 0.162, $t(19) = -2.366$, $p = 0.029$). These results suggest that the temporal performance decline occurred in cooperative mode regardless of sitting arrangements or group roles, but not in the competitive mode.

2.3.6 Behavioral change after making an error

While examining the reaction time for each answer during gameplay, an interesting observation was made regarding players response immediately after making an erroneous selection in the Stroop test. A typical response is shown in Figure 2.5, where the reaction time to the next question immediately after an error (indicated

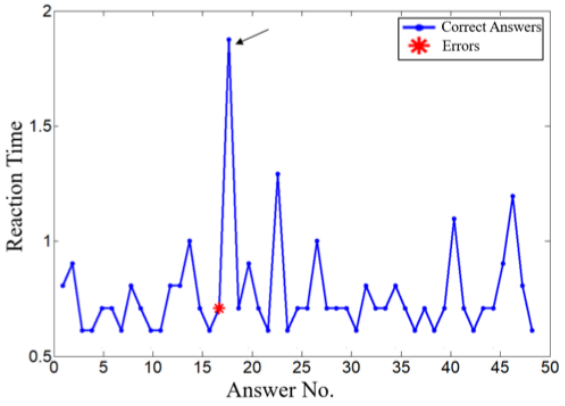


FIGURE 2.5: A players reaction times during a round of gameplay. Note the significant increase in reaction time immediately after an error (red asterisk).

by a red asterisk) suddenly increases. Based on this observation we are interested to answer the following research questions:

RQ6: Will making an error influence players subsequent gameplay behavior? And if so, how does this influence differ in the different gameplay modes (competitive vs. cooperative) and different sitting arrangements (close vs. apart)?

The average reaction times immediately after an erroneous selection were labelled as post-error reaction times, while the average reaction times immediately after a correct selection were regarded as the normal reaction times. The data of the participants who did not make any mistake were excluded in the analysis. A two-factor (post-error gameplay) repeated measures ANOVA and a two-factor (post-error sitting arrangement) repeated measures ANOVA were conducted for the reaction time. Main effects of post-error were found in both studies ($F(1, 20) = 15.747$, $p = 0.001$, $F(1, 18) = 26.200$, $p < 0.001$) respectively. However, no main effect of gameplay modes or sitting arrangements or any interactions were observed (all $p > 0.1$). Follow-up paired t-test suggests that the post-error reaction times were significantly longer than normal reaction time in all three gameplay modes. Specifically, in the competitive mode, the post-error reaction times are significantly longer ($M = 1.390$, $SD = 0.553$) than the normal reaction time ($M = 1.068$, $SD = 0.213$), ($t(32) = 3.724$, $p < 0.001$). Also, the post-error reaction times for cooperative mode - apart and cooperative mode - close of ($M = 1.237$, $SD = 0.190$) and ($M = 1.223$, $SD = 0.279$) respectively were also significantly degraded from the normal reaction times of ($M = 1.068$, $SD = 0.123$), ($t(22) = 4.851$, $p < 0.001$) and ($M = 1.060$, $SD = 0.162$), ($t(27) = 3.259$, $p = 0.003$). These results imply

that a players reaction time tend to slow down immediately after making an error and this is unaffected by whichever gameplay modes or sitting arrangements.

Our cooperative mode design adopted strong peer accountability which will cause the error made by one player to affect the score the other can receive. In addition, this partners error is made explicit through audio and visual feedback on the tablet computer. As such, we are interested to explore the effects of social context and asked the following research question:

RQ7: Will making an error influence the partners subsequent gameplay behavior during cooperation? And if so, how does this influence differ with the different sitting arrangements (close vs. apart)?

A two-factor (sitting arrangement post-partner-error) repeated measures ANOVA was conducted for the reaction time. There was a main effect of post-partner-error ($F(1, 18) = 5.450$, $p = 0.031$) found for reaction time, but no main effect of sitting arrangement ($F(1, 18) = 1.197$, $p = 0.288$) or interaction effect ($F(1, 18) = 1.402$, $p = 0.252$) were observed. This results suggest that during cooperation the partners error can indeed affect the players behavior. Follow-up paired t-test revealed that in cooperative mode apart there was no significant difference between the post-partner-error reaction times ($M = 1.181$, $SD = 0.372$) and the normal reaction times ($M = 1.051$, $SD = 0.122$), ($t(22) = 1.628$, $p = 0.117$). However, in cooperative mode close, the reaction times were found to significantly degrade after the partners erroneous responses ($M = 1.171$, $SD = 0.353$) compared to normal reaction times ($M = 1.040$, $SD = 0.169$), ($t(27) = 2.698$, $p = 0.012$). These findings imply that even though the reaction times were not observed to be affected by a partners error when cooperating from a distance, this cannot be said when they were cooperating in close proximity. Making a mistake seated beside a partner not only causes ones reaction time to slow down in the next attempt, it also causes ones team mate who did not make any mistake to adopt a similar cautious behavior.

2.3.7 Discussion

This study examined the influence of peer accountability on players' gameplay behavior, in particular their attention levels. The results show that players made

significantly fewer errors when they were cooperating than when they were competing against each other, suggesting that the sense of peer accountability during the cooperative gameplay has improved their focus and attention in the cognitive task. However, these results differ from those of [46], where they observed no significant performance differences between cooperative and competitive goal structures. There are several major differences between these two studies that can help understand the appropriate context in which the findings in this study can be generalized.

The first is the nature of the gameplay task. The impact of cooperation varies with the type of task employed. The study in [46] used a motor activity-centered task requiring players to pop as many balloons as possible on a computer screen within a given time. Instead of physical activity, the study in this work employed a cognitive task that requires players to focus on a stimulus and make a corresponding multiple-choice selection, and repeat this correctly as many times as possible within a given time. In addition, this study used the simple Stroop Color-Word Interference test and tertiary-level student participants in order to reduce the influence of task competency and content knowledge on the players' ability to perform the cognitive task. In this way, the performance is predominantly based on the players' ability to focus their attention during gameplay. Note that using complex cognitive tasks such as those involving arithmetic skills have resulted in contrary results which show competitive and not cooperative mode producing increased game performance relative to individual gameplay [64]. In short, this study adds to previous research findings by showing that cooperation can also benefit cognitive tasks that require focus and selective attention.

The second is the extent of the peer accountability incorporated into the cooperative game design. This study emphasizes explicit and strong peer accountability by ensuring one's progress in the game is peer dependent and all players' current performance is mutually visible, as illustrated in Figure 2.1(b). In this conjunctive design of the cooperative play, the participants were made aware that they were simultaneously attending to the same task as their partner. Based on the shared attention theory, this conjunctive design can evoke awareness of shared attention as people tend to devote more cognitive resources to tasks that are thought to be synchronously co-attended with another [89, 90]. In contrast, the cooperative

game design of the balloon popping game [46] uses implicit or weak peer accountability as the individual's effort during the cooperative gameplay is independent from the partners'. Hence, the results presented with the conjunctive Stroop test only suggest that performance improvement in terms of reducing error rates during cooperation may be applicable to cooperative designs with strong peer accountability. It remains to be investigated if similar positive effects of cooperation can be observed in the scenarios with the weak peer accountability, such as those in [46].

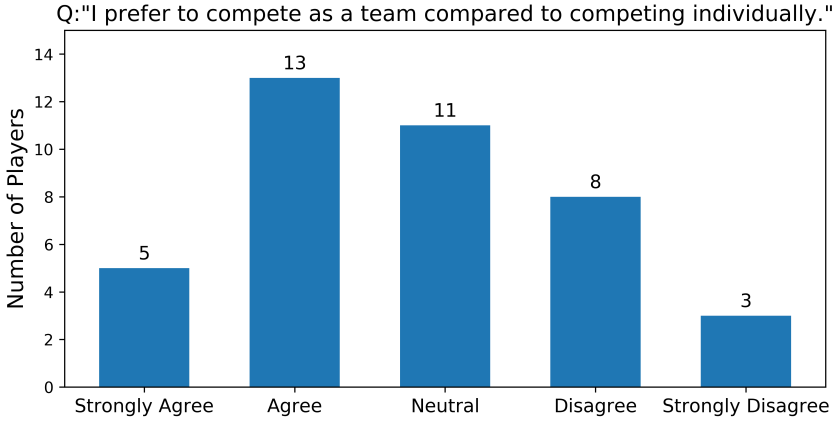


FIGURE 2.6: Qualitative results from a 5-point Likert scale survey question asking players to compare their preference for the two gameplay modes. The numbers selecting the respective response are indicated above each bar plot.

There are several interesting findings in this study that deserves further discussion. First, this study observed that more accurate performance does not necessarily translate to a more positive gameplay experience. Players' preferences for the two gameplay modes compiled from the questionnaire survey show a more mixed result, as shown in Figure 2.6. Although there were more players preferring the cooperative mode ($N = 18$) than the competitive mode ($N = 11$), there were still 11 players who held neutral feelings. A sampling of comments from participants may provide further insights. Some players expressed they were motivated by the responsibility to the partners during cooperation, such as "*Competing as a team provokes my desire to win for the consideration of other team members*", "*I felt more responsibility to win the game as a part of team*", "*Competing as a team makes me feel greater sense of confidence and I hold the feeling of not wanting to disappoint the other member*". However, other players alluded to the loss of control and the increased stress introduced by the peer accountability "*Playing as individual is easier*", "*The response of the game to my answer is not as timely as*

that in competing individually", "*Competing as a team increases the responsibility greatly, along with it I feel more stressful. I prefer to work individually*". These mixed preferences could be the result of the strong peer accountability employed in the game design. This can make players feel more motivated because of the strong sense of interdependency and teamwork. But it can also make some feel more nervous and conscientious about their own performance. Field dependent/field independent (FD/FI) cognitive styles of each individual may also explain the mixed preferences observed in this study. The FD/FI dimensions [91] do categories people into those that prefer group and cooperative activities (i.e. FD) and those who prefer individualistic and competitive activities (i.e. FI).

Another interesting observation is that peer accountability appears to have a more positive influence on the performance of the slower players than the faster players. The slower players made significantly fewer errors when cooperating than when competing with their faster partners. The fact that these slower players can further improve their accuracy during cooperation suggests that being accountable to a superior or faster partner can create a strong incentive to put in more effort and focus. This observed phenomenon is reminiscent of Lev Vygotsky's view that interaction with more skilled peers is an effective way for less competent learners to develop skills and mastery [28]. Even though his theories are related to social learning, the benefits of cooperation with a stronger peer seem to be applicable to multiplayer gameplay too. More pertinently, the findings in this study mirror that of the Köhler motivation gain effect, first described in the 1920s by Otto Köhler, a German industrial psychologist [92]. He observed that the weaker member of a group will exert extra effort that is beyond the usual performance limits when paired with a stronger partner in a conjunctive task. Such tasks are similar to the cooperative task with strong peer accountability used in this study, where the group's productivity or performance is equal to that of the weakest member. Several social-psychological mechanisms have been suggested to explain this phenomenon [93]. One is the social comparison process that encourages one's personal performance goals to be revised upwards when working with a more capable partner as one becomes aware of a higher performance standard. Another is the player's sense of indispensability to the group. The more indispensable one perceives one's effort is to the group's outcome, the greater the effort one will exert. This is normally associated with the feelings of not wanting to let the team down. It is interesting to note that most studies related to the Köhler motivation gain

effect have been described in the context of physical activities such a rowing, tethered mountain climbing and exergaming [94–96]. However, this study has provided empirical support that this effect is equally applicable to a cognitive conjunctive task. This implies that the Köhler motivation gain effect can also be employed to motivate and pull up the "weaker link" in cooperative activities requiring focus and attention, not just physical effort.

In addition, the dynamics of cooperation between unequal partners appears to be more complicated than it first appears. The influence of peer accountability on the player's behavior is not always positive. The analysis of average in-game reaction time data shows that cooperating partners who have highly mismatched competencies are unable to sustain their performance. There was significant degradation in their average reaction time by the end of the gameplay duration but this temporal slowdown was not frequently observed with the closely-matched pairs. Similar negative side effects in cooperation were observed by researchers studying the impact of team ability disparity playing a Counter-Strike game [49]. They found that under a cooperative reward structure, playing in a team with high ability disparity had a negative influence on individual performance. Interestingly, Köhler also observed that the amount of motivation gains is dependent on the extent of discrepancy between partners' abilities [92]. The Köhler discrepancy effect suggests that there is an optimal ability disparity that can encourage the weaker player to improve but when this disparity is too large, the motivation gain will start to decrease. The sense of indispensability when presented in a highly mismatched group can create high stress and in turn affect sustained performance. This was indeed reflected by some of the comments in the questionnaire survey such as "*Competing as a team gave me more pressure. I can't do anything wrong as I need to be responsible for my partner*". It is unclear if this observed behavior can be generalized to other types of cooperative activity designs such as those that have weaker peer accountability or when the ability disparity is measured by other dimensions besides average reaction time. Nonetheless, this finding provides further support of the relevance of the Köhler discrepancy effect in the attention-based cognitive task and its implications on how optimal teams should be formed. Where possible, it is a prudent strategy to ensure the abilities of team members are not severely mismatched if one hopes to sustain good teamwork performance.

Lastly, there are numerous studies on the social and cognitive affordances of spatial features such as distance, proxemics, co-presence and physical visibility of shared context [74]. The influence of close proximity has been associated with positive emotional, cognitive and behavioral changes in work groups [97]. However, this study comparing sitting arrangements during the cooperative gameplay shows that physical proximity does not have any significant positive influence on players' performance when they are cooperating over a task that required each individual's sustained attention and cognitive focus. This result is not surprising when we consider the fact that the cooperative Stroop test task used in this study can be accomplished with minimal communication and consultation between partners. As such, it is not representative of typical cooperative tasks where the positive effects of conversation [98], task and group awareness [99] can be facilitated by physical proximity. From the results of the user survey, it appears that people are psychologically affected in different ways when they cooperate in close proximity. Some players were affected positively and shared comments such as "*I was more focused because my teammate was sitting beside me" and "when sitting close, the sense of working in a team is stronger*". However, others felt increased pressure and distraction. They shared that "*sitting close brings pressure*" and "*sitting further from my partner decreases the distraction, thus I was more focused*". In short, it has been observed that when cooperating over a task that required focused attention, close physical proximity did not always bring positive influence but might create undesirable distractions.

2.3.8 Limitations and Future Work

There are several limitations in the generalizability of the findings in this work. Most notably, the simple cognitive-oriented Stroop task used in the multiplayer gameplay design is not representative of typical video games, which usually incorporate many complex gameplay and cooperative elements. This study is predominantly focused on the influence of peer accountability, which is most representative of the commonly used cooperative design pattern called complementarity [48, 100]. It refers to the mechanism that gives players the abilities to complement each other's activities. In the game design of this study, the left lock complementing with the right lock determines whether the players can successfully get the point and move to the next question. Similar applications of such design feature can be

found in First Person Shooters (FPS) games. For example, to reach higher grounds, players need to come to the spot together and piggyback on each other [100]. In this case, the progress of the team is also dependent on cooperative contribution. Hence, the results of this study might shed some light on the dynamics between faster and slower players as well as the influence of their performance disparity in this specific case. However, besides complementarity, many other cooperative design features are often employed and intertwined with each other in typical video games. For example, video games with more complex character designs can support synergies between abilities [48, 100] to allow one character type to assist or change the abilities of another. The findings in this study may not be generalizable to such complex multiplayer game design scenarios. Further research is needed to study how different cooperative design patterns and the combination of them may affect players' attention in more realistic video game settings. The challenge of such an endeavor is how to account and assign the measured influence to the different cooperative design elements as mutual interactions are hard to avoid.

In addition, a simple cognitive task in the form of a Stroop test was employed in order to measure the attention levels using basic quantitative measures such as response time and error rate. However, real-life cooperative applications involving problem-solving and complex decision making often require higher levels of cognitive skills. Under such scenarios, players could exhibit very different interaction dynamics and cognitive states during competition and cooperation. Studies employing more challenging cognitive tasks such as multiple-choice-multiple-answer questions and more sophisticated measures of player performance would be needed to further explore this comparative investigation using higher cognitive skill sets.

2.4 Summary

This study presented quantitative empirical support for the positive influence of peer accountability on attention levels in a conjunctive cognitive task. Further evidence was also provided to support the Köhler motivation gain and Köhler discrepancy effects by demonstrating that the Köhler effects are equally applicable to cognitive-oriented cooperative tasks as they are to physically-oriented ones. The implication of these findings for games or collaborative application designers is that one can employ a cooperative design with strong peer accountability to increase

players' attention and performance in terms of accuracy. However, to maintain such positive influence of peer accountability, the discrepancy between the members' performance must be properly moderated. Such performance disparity among players can be ameliorated if there are some meaningfully ways to match or level up the difficulty of the task presented to each individual player based his or her ability. This challenge of task difficulty adaptation is the focus of the remaining chapters of this thesis.

Chapter 3

Difficulty Adaptation for Visual Memory Game

3.1 Background

In the previous chapter, a user study was conducted to explore design factors that might influence users' attention during gameplay. A key observation in this study is that there is a significant performance decline when the performance disparity between the gaming partners is large. In a conjunctive cooperative task, if one player finishes the task much faster than the other partner, the faster partner needs to wait a long time for the other to keep up. This waiting event can make the slower one feels anxious and the faster one feel bored. This may lead to overall performance decline among both players. Therefore, to maintain high learning gains, the performance disparity among group members needs to be moderated. One way to reduce the performance disparity is to present faster players with harder tasks and the slower players with easier ones. In other words, the task difficulty should be matched with the player's ability. This leads to the problem of *dynamic difficulty adaption* (DDA).

DDA has important applications in multi-disciplinary fields including education, gameplay, etc. [9, 15, 17, 101–103]. Additionally, besides the applications in the multiplayer setting as discussed in Chapter 2, previous studies have established that difficulty adaptation can also greatly benefit individual settings. In the psychology literature, the flow theory [27] suggests when the challenge matches with

user ability, the user will operate in a flow zone where that the concentration and engagement can be enhanced to the optimal level. On the other hand, if the task is too hard or too easy, it will break the flow and may result in anxiety and boredom respectively. Besides theoretical support, there are also some empirical studies, in which DDA has been used to tailor game based individual needs and generate engaging experience in computer games [15, 17, 101], and to increase motivation and scaffold learning in intelligent tutoring system [9, 102, 103]. Specifically, Hunicke and Chapman [17] applied difficulty adjustment in a First Person Shooter (FPS) game to keep player in the flow channel, by detecting whether the player is repeatedly in trouble and provide the necessary resources to help them get out of it. Liu et al. [15] proposed a difficulty adaptation mechanism based on player affective state, which uses physiological signals, obtained via wearable biofeedback sensors, to predict player anxiety level and then adjust the game difficulty accordingly based on deterministic rules. On the educational front, to investigate the effectiveness of adaptive difficulty adjustments, Sampayo-Vargas et al. [9] conducted a user study with 234 secondary school students learning Spanish cognates. The pre and post-tests results showed that the learning outcomes of the group with the adaptive game are significantly higher than that of the non-adaptive group, which suggests the difficulty adaptation can provide a scaffold structure to facilitate learning.

In short, the problem of dynamic difficulty adaption is an important area in improving the design of interactive systems and maintaining user performance. This is the focus of the remaining chapters of the thesis.

3.2 Stimuli

3.2.1 Visual Memory

To study DDA problem, a visual memory task is chosen as the stimuli. Working memory (WM) refers to the ability to hold information during short time periods [104]. It has been argued that the short-term storage and manipulation of information in memory is one of the key components in cognitive activity [105]. The working memory capacity, which indicates the maximum amount of information that can be retained in the working memory, is an important factor impacting the ability for problem solving and reasoning [104]. In fact, a correlation has been

found between the individual differences in working memory capacity and the differences in academic achievement [106].

Traditionally, working memory capacity is believed to be constant. However, more recent research works suggest that working memory can be enhanced by training. For example, Kingberg et al. [107] conducted a controlled trial with 53 children who suffered from attention-deficit/hyperactivity disorder (ADHD). After conducting computerized memory training with more than 20 days, the performance on the testing visuospatial task (span-board task) was significantly improved in both the post-test and the follow-up test after 3 months. The results suggest that the working memory can be improved after the computerized, systematic practice of WM tasks. In fact, a meta-analysis of 25 studies of cognitive training for children with ADHD revealed that computer-based cognitive short-term memory training can result in improvements in short-term memory [108]. Moreover, the effectiveness of working memory training does not only exist for ADHD patients, but some studies also observed such improvement among healthy subjects. Specifically, after training with a visuospatial working memory task (a letter span task and a backward digit span task) for 5 weeks, the FMRI results for healthy adult human subjects showed that neural activities in prefrontal and parietal regions were increased [104]. Such changes in cortical activity seem to suggest the memory training can lead to some level of plasticity in the neural systems that control the working memory. In summary, multiple empirical evidence indicates that WM training program can lead to significant improvements in general WM capacity. However, it should be noted that it does not mean all the WM training programs lead to positive effects. To achieve successful results, it is necessary to make the training intense and adaptive. As a matter of fact, no substantial gains in working memory were observed in non-adaptive training where the difficulty was always set to the initial low level and sustained memory improvements were only observer in adaptive version training that matched task difficulty with child's memory span [109]. Therefore, to ensure the effectiveness of memory training, it is necessary to make the task difficulty adaptive to users.

To achieve difficulty matching between the visual memory task difficulty and the player's ability, we first need to determine the difficulty of a task for a user. Previous studies usually used handcrafted rules to determine task difficulty. For instance, in the commonly used memory span task, in which the subjects need to remember

and recall a series of ordered items, the number of the items is often regarded as the difficulty indicator. Depending on the items employed in the span task, there are digit span task, letter span task, and spatial span task, etc. In these memory span tasks, the length of items is a key factor for information load and thus a reasonable measure for the memorization difficulty. However, for some other memory tasks, the determination of memorization difficulty is not so straightforward. One example is the visual memory task. The capacity of visual working memory is influenced not only by the number of items but also by the visual information loads of each item, such as the features of color, orientation and conjunctions [105, 110–112]. Furthermore, other research works have shown that the visual recall of a specific item can be influenced by the other proximal items in the visual scene. Specifically, perceptual grouping can facilitate visual working memory and allow factors such as adjacency, alignment and compound shapes to aid memorization [113]. In fact, Brady et al..[114] argued "*every display has multiple levels of structure, from the level of feature representations to individual items to the level of groups or ensembles, and these levels of structure interact*". To highlight the complexity of visual memory, a spatial visual memory task is used in our study (see Figure 3.1). In

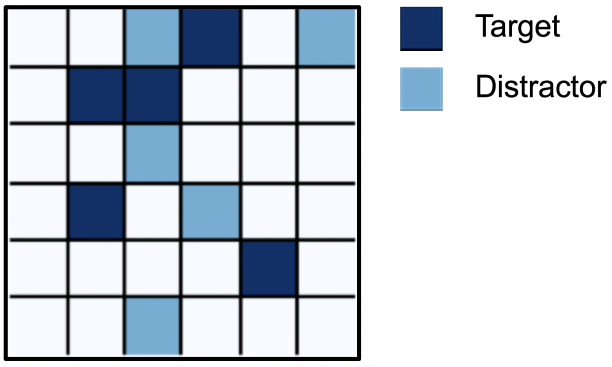


FIGURE 3.1: Spatial-board task: targets are shown simultaneously.

this task, all the targets are *simultaneously* shown to the subjects. The task is to memorize and recall the positions of all the targets. This spatial-board task is chosen as stimuli because the difficulty of such a visual memory task cannot be easily decided by handcrafted rules, like the number of targets. For instance, Figure 3.2 shows two tasks both with 4 targets. The rules based on the number of targets will label them with similar memorization difficulty. However, most people will find the Task No.16 with the scattered targets more difficult to remember and recall than Task No.18 that contains clustered structure. Therefore, a simple rule-based ranking can be unreliable.

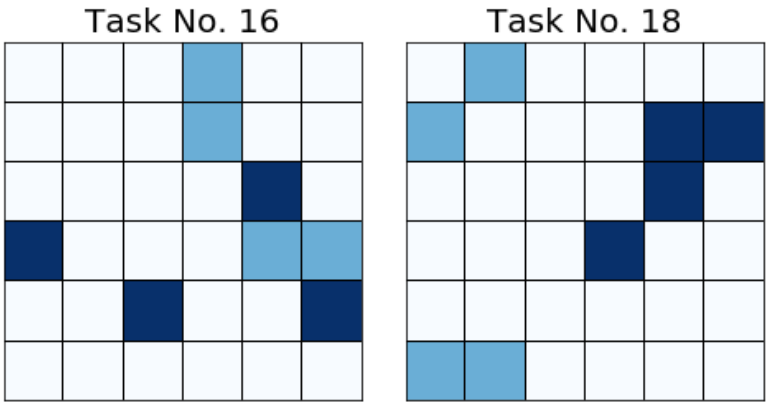


FIGURE 3.2: Two examples of 4-target visual memory tasks: Task No.16 and Task No.18.

Moreover, when the question bank becomes large, the rule-based difficulty ranking can be difficult to formulate reliably. Take the two visual memory examples in Figure 3.3. The Task No.16 contains only 4 targets and the Task No.90 contains 8

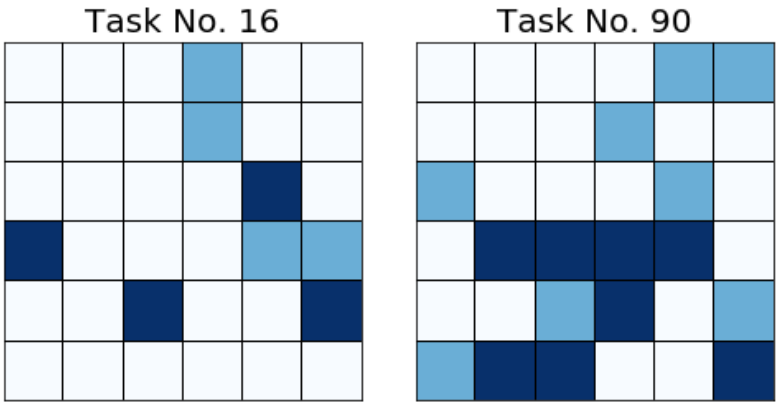


FIGURE 3.3: Two examples of visual memory tasks: (a) a task with 4 targets; (b) a task with 8 targets.

targets. But this does not necessarily make the Task No.90 harder to memorize as the targets form distinctive structures. In fact, the relative difficulty of each task may even vary from person to person based on the individual's visual memory characteristics and memorization strategy. For instance, people with a good perception of connected structures may find it easier to memorize the Task No.90 despite it contains more targets. This is called the problem of *difficulty ranking personalization* (DRP). It should be noted that DRP is not a unique problem for visual memory game alone. It has been a recurring issue in e-learning design. Based on

students' different backgrounds and styles, the relative difficulties of the questions are also different [115]. With the increasing adoption of online education platform among a diverse user base, there is a growing need to accommodate individual differences using some form of difficulty ranking personalization.

3.2.2 Game Design

Based on the spatial-board idea mentioned earlier, an online game, called *Pals*, was designed as the test bed for the DDA study.

The game was designed with several goals in mind. First, the game needs to incorporate the visual memory task as part of the gameplay. Second, since the central concern in the DDA research is related to how hard the visual memorization task is to the user, a quantitative way to measure the task difficulty must be designed, preferably via in-game performance data. Third, as the visual memory task generally requires a high level of sustained attention, the game design needs to motivate the players to be focused and engaged with the gameplay till the end of the game. The remaining part of this section explains how these design goals were addressed in the game design.

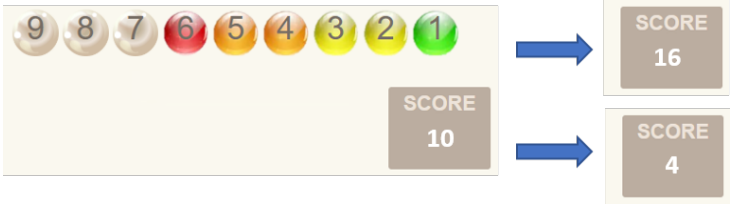
3.2.2.1 Gameplay

The spatial-board task is embedded in the gameplay in the following way. In the game, the player needs to complete 25 memorization tasks correctly. For each task, a formation, which consists of several targets and distractors, is shown to the player (see Figure 3.4) after the "Start" button is pressed. The goal of the player is to memorize the positions of all the targets as quickly as possible and recall it correctly. The players decide how much time is spent on the memorization of each task. When the players feel ready, they press the "Ready" button to proceed to recall the formation. During the recall stage, the formation, including all the targets and distractors, will disappear (see Figure 3.4). Player needs to recall the positions of the targets by clicking on the grids. After the player places all the targets, the results will be shown in green for correct placement and in red for incorrect ones.

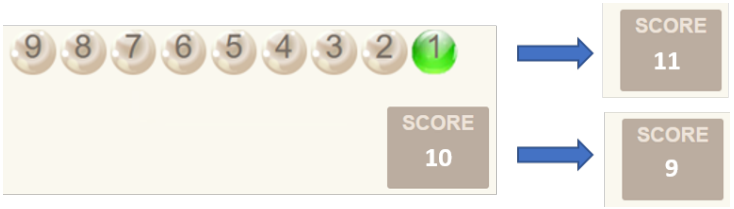
FIGURE 3.4: An example of a gameplay sequence in the Pals visual memory game triggered by the press of the buttons Start and Ready. The Recall stage ends when the user has placed the required number of targets.

3.2.2.2 Scoring Mechanism

The scoring mechanism is as follows. If the recall is perfect, which means the positions of all the targets are correctly identified, the player will earn some points. The less time the player takes on the memorization, the more points will be given. However, if the recall is wrong, i.e. one or more of the placements are incorrect, the player will lose points instead. Specifically, the memorization time is indicated by time bubbles. The increase and decrease of game score is determined by time bubbles. At the start of each task, there are 9 colored bubbles. During the memorization stage, the time bubbles will grey-out one by one as time elapses, until the memorization stage ends (or there is only one bubble left). And the number of bubbles left at the end of memorization stage is used to decide the game score. For example, in Figure 3.5 (a), the memorization takes three bubbles and there are six bubbles left. Thus, the score will be increased by 6 if correct and decreased by 6 if wrong. In 3.5 (b), when the user is slower in memorizing and takes up 8



(a) After memorization, there are 6 bubbles left.



(b) After memorization, there is 1 bubble left.

FIGURE 3.5: Time bubble and score mechanism:(a) score increased/decrease by 6; (b) score increased/decrease by 1;

bubbles, the score will only increase (or decrease) by 1, the only remaining bubble. The relationship between the memorization time and the number of bubbles left (i.e. the change in score) is shown in Table 3.1. To accommodate players' different memorization ability, the time bubble disappears in a non-linear speed. The first three bubbles disappear at every 800 ms and the next two at every 1000 ms and the next two at every 1200 ms. The last one does not disappear. The use of a nonlinear relationship between time bubbles and memorization time is to better accommodate different memorizing abilities of users. The first few bubbles disappear at a fast speed to make the players with very good memory still feel challenging. The last few bubbles disappear at a slow speed to make sure the bubbles do not disappear too early for the players with poor memory.

The time spent on memorization is used as the quantitative measure of the task difficulty in the data analysis. The harder a task is, the more time will be needed for a user to memorize it correctly. However, recall that in the game, the memorization time is actually decided by the users themselves. Some users may be overcautious and take more time than they really need to memorize the pattern. On the other hand, some may make reckless guesses to complete the task as fast as possible. The proposed scoring mechanism is specially designed to try to discourage these two scenarios and this is based on the assumption that players will behave in a manner that will maximize their score. Reckless guessing behavior is discouraged using score penalty during incorrect placements and focused best-effort memorization is

TABLE 3.1: The relation between time bubble number and the corresponding memorization time.

Time Bubble No.	Memorization Time (ms)
9	<800
8	800-1600
7	1600-2400
6	2400-3200
5	3200-4200
4	4200-5200
3	5200-6400
2	6400-7600
1	>7600

encouraged by rewarding higher scores for shorter memorization time. In summary, this scoring mechanism is designed to ensure the memorization time can more accurately reflect the task difficulty for the user.

3.2.2.3 Motivation Design

To increase the playfulness of the game, a story narrative is embedded in the reward animation. The goal of completing 25 memorization tasks correctly can be interpreted as collecting 5 planks. To collect one plank, the player need to complete five visual memory tasks correctly. At the completion of each task, a cute animation is shown to the player which displays an action conducted by two "pals". The five actions involved in the collection of a plank are: conveying the plank by panda and bear (see Figure 3.6(a)), sawing the plank by fox and squirrel (see Figure 3.6(b)), pulling the rope by rabbit and hedgehog (see Figure 3.6(c)), conveying the plank further by panda and bear (see Figure 3.6(d)), and pulling the rope higher by rabbit and hedgehog (see Figure 3.6(e)). The five actions are logically connected. After every five actions, a plank will be successfully collected and shown in the plank collection bar.

At the end of the whole game, there will five planks in the cube (see Figure 3.7). This plank collection design aims to motivate players to finish the whole game of 25 tasks. After the player finished the whole game of 25 tasks, the score and ranking information will be presented (see Figure 3.8). The nearest scores in the database will also be displayed, with the goal of encouraging the players to better

(a) (b) (c) (d) (e)

FIGURE 3.6: Five cute animations used to motivate sustained gameplay: (a) convey the plank; (b) saw plank; (c) pull the rope; (d) convey the plank further; and (e) pull the rope further.

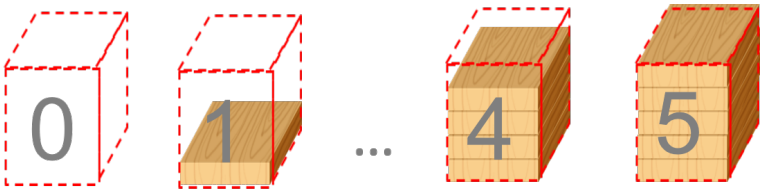


FIGURE 3.7: Plank collection bar which can contains five planks.

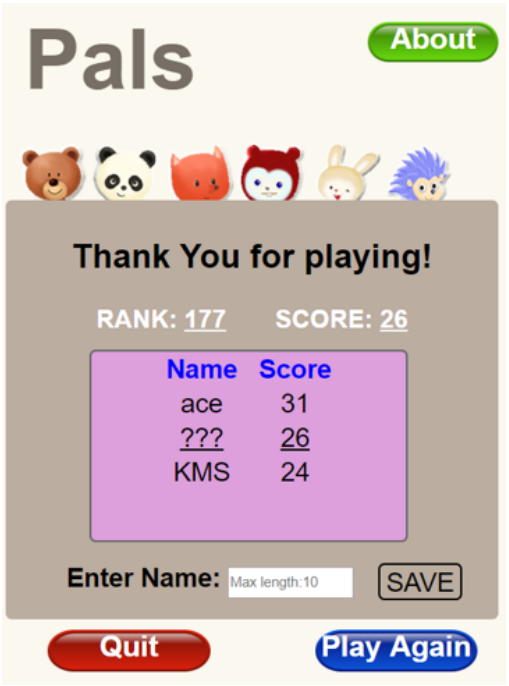
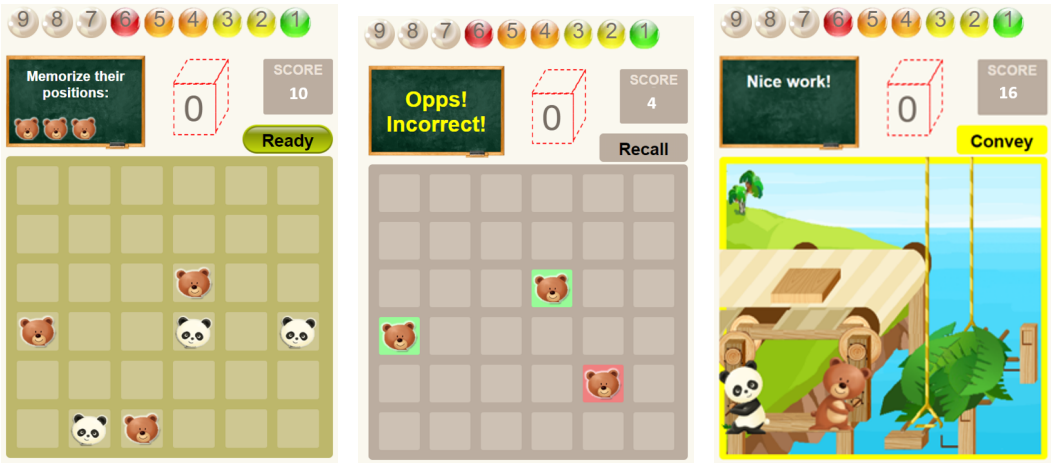


FIGURE 3.8: Final scene at the end of the game with the game score and ranking info shown.

their position in the scoreboard by playing another round of 25 tasks. The player can enter their own nicknames and their names will be shown on the ranking board

to generate a sense of achievement for the players. Repeated play is encouraged so that the online visual memory game can collect more performance data.

The full game scene with the span-board, time bubble, plank collection bar together is shown in Figure 3.9.



(a) (b) (c)

FIGURE 3.9: Full game design: (a) A visual memory task is shown to the user to memorize; (b) After recall, the result is shown with correct placements in green and wrong ones in red. (c) For correct answer (i.e. all placements are right), an animation is shown.

3.3 Experimental Results

3.3.1 Visual Memory Game Implementation

The game was implemented in Javascript, CSS, PHP and published on a website[1] supported by Amazon Web Service (AWS). The game data is stored in Relational Database Service on the AWS.

The task set consists of 100 visual memory tasks in total, which were randomly generated. The targets number lies between 3 to 8. The full question bank can be found in the Appendix A.2. In the preliminary study, the task was randomly chosen from the task set at each time step. The participants were recruited from the Amazon Mechanical Turk platform. 77 subjects participated in this study. The

[1] *http://vmg23apr-env.wipf9rh8mt.ap-southeast-1.elasticbeanstalk.com/vmg_23_Apr/*

data collection was conducted after the approval of the NTU-Institutional Review Board (IRB-2018-06-029).

3.3.2 Terminology

The task set is denoted as $\mathcal{A} = \{a_1, a_2, ..., a_A\}$. The difficulty level of a task a_j is denoted as $D(a_j)$. Difficulty ranking on a task set is described by a vector, which includes the difficulty levels of all the tasks in the task set $DR = [D(a_1), D(a_2), ..., D(a_A)]$.

The player set is denoted as $\mathcal{P} = \{p_1, p_2, ..., p_N\}$. Given a task a_i and a player p_j, the perceived difficulty level of this task for the player is denoted as $D_j(a_i)$, which is measured by the time that the player spends on memorizing the task. The difficulty ranking of the player p_j is denoted as $DR_j = [D_j(a_1), D_j(a_2), ..., D_j(a_A)]$

3.3.3 Preliminary Results

This preliminary study aims to investigate whether there is a *universal* difficulty ranking for all the players. To this end, we examined the relative difficulties of question pairs for each player. In other words, given two questions, we studied whether people have a common idea on which one is harder.

The results suggest that people can differ in which task is harder and which task is easier. For example, Figure 3.10 shows two tasks, 55% of players felt Task No.16 harder, i.e. deserves higher memorization time, and 45% of the players felt Task No.49 harder.

Similarly, for the two tasks shown in Figure 3.11, 42% of people found Task No.49 was harder while 58% found Task No.77 to be harder. These results seem to imply people cannot agree on the relative difficulty of the question pair. In fact, among the 1475 question pairs [2] that were examined, this phenomenon exists for more than one third (34.2%) of them, with about half (40% - 60%) of players indicating one is harder and the other half (60% - 40%) indicating the opposite. In summary, it seems that there are more than one difficulty ranking for visual memory tasks and

[2]The condition for selection of question pairs is that both questions need to be played by more than 4 players. Among the data collected, there are 1475 question pairs meeting this condition.

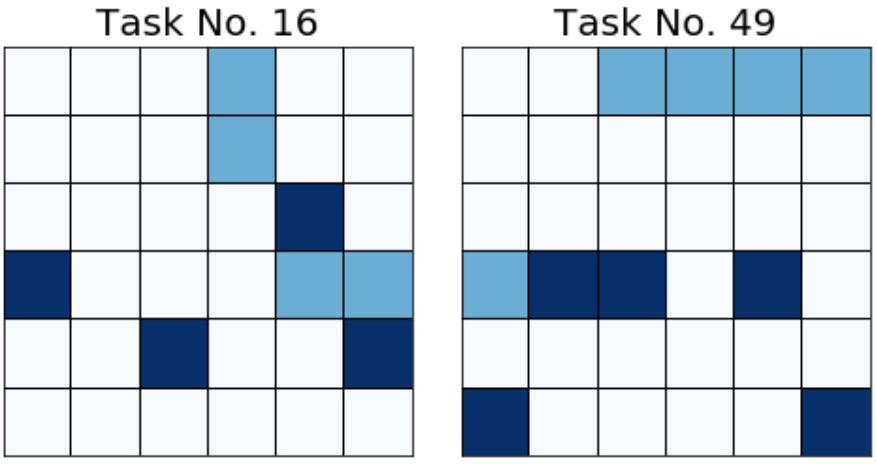


FIGURE 3.10: A question pair with Task No.16 and Task No.49

a single universal difficulty ranking will not be representative of the way different users memorize a given visual memory task.

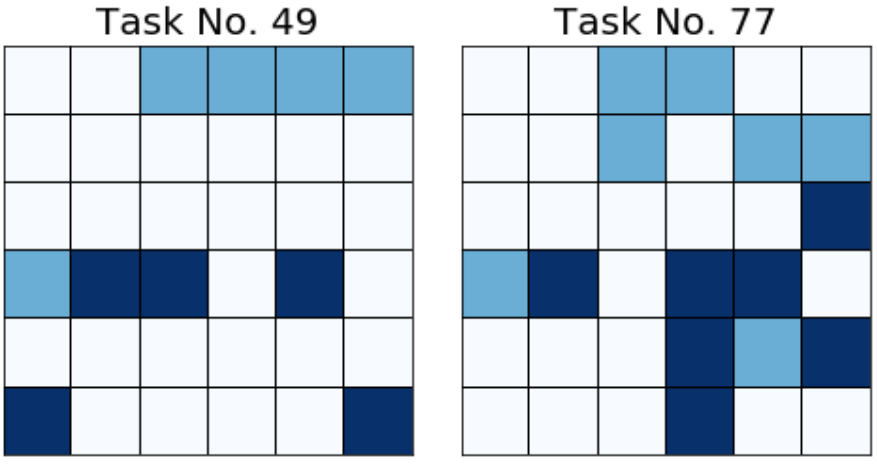


FIGURE 3.11: A question pair with Task No.49 and Task No.77.

3.4 Summary

Motivated by the findings of Chapter 2, dynamic difficulty adaptation (DDA) is proposed as an important area to improve interactive system design. DDA aims to match the task difficulty with user ability by presenting the strong users with hard tasks and the weak users with easier ones. In order to embark on the DDA study, a spatial visual memory task was chosen. Its complex and user-variable difficulty

characteristics is representative of typical task difficulty encountered in many applications and will make the proposed DDA investigations more generalizable. An online game platform was specially designed to embody the visual memory task and quantitatively capture the task difficulty through in-game performance data in the form of user's task memorization time. The preliminary study on the visual memory game platform confirms the highlighted challenge in DDA, which is that different people exhibit different expression of which tasks are hard to remember and which ones are easy. The next chapter will deal with this challenge and discuss how the personalized difficulty ranking for a given user can be obtained.

Chapter 4

Clustering-based Difficulty Ranking Personalization

As discussed in Chapter 3, each person has a different sense of which visual memory task is easier or harder than which. This variation in the order of the task based on it increasing difficulty is termed *difficulty ranking*. The difficulty ranking for a user reflects the characteristic of the user's strength and weakness in performing the series of different tasks and is termed individual's *difficulty ranking profile* or *difficulty profile* for short. The process to identify the difficulty ranking profile of each user is called *difficulty ranking personalization* and is the focus in this chapter [1]. Specifically, this chapter concerns the problem as shown in Figure 4.1. Given a

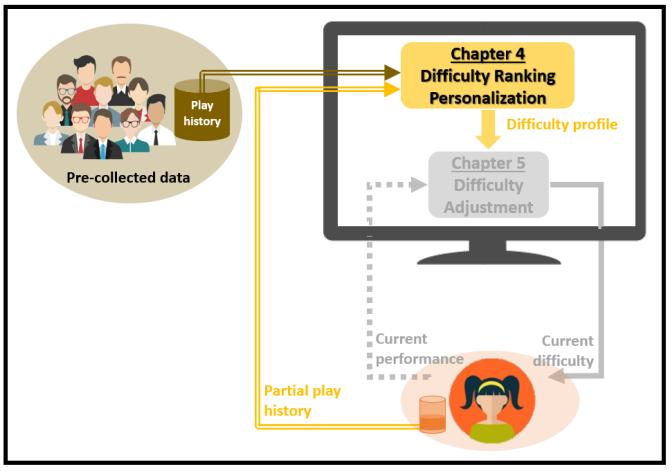


FIGURE 4.1: Problem structure of difficulty ranking personalization.

[1]The section 4.1 in Chapter 4 is published as [116].

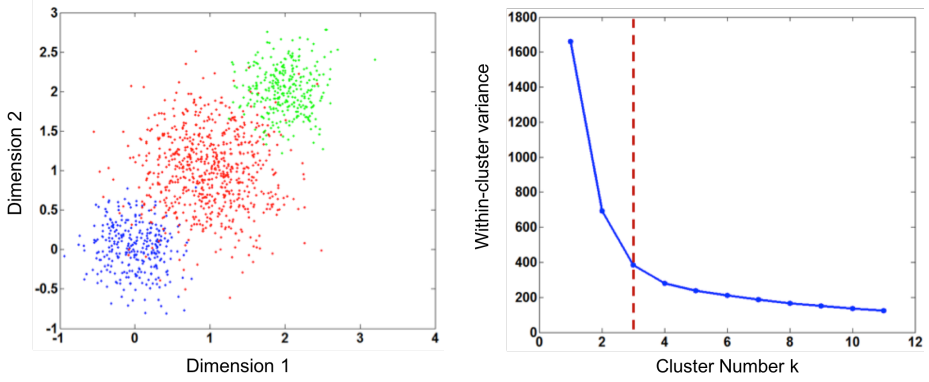
new user, the system needs to predict the difficulty ranking profile for this user on the fly based on his or her partial play history while interacting with the system.

To learn personalized difficulty rankings for each user, previous method [115] proposed a KNN style method by combining collaborative filtering algorithms with social choice theory. During the testing stage, this method needs to compare the distance of the new sample with all the samples in the training data to determine the neighbors. This high computational cost is acceptable for offline scenarios but is impractical for real-time responsive applications. To overcome this issue, this work proposes to first build a prototype among the training data via clustering, then compare the distance with the cluster centers. However, to perform clustering, it is unclear how many different distinctive visual memory difficulty profiles exist or can be discriminated. The preliminary results in Chapter 3 only indicate there are more than one kinds of visual memory difficulty profile among users. To solve this problem, Section 4.1 presents a technique to determine the likely number of distinguishable difficulty profiles in a sampled population. Based on this technique, Section 4.2 proposes a clustering-based difficulty personalization method and applied it in an online visual memory game platform. Examination of the results of difficulty ranking personalization produced some interesting insights regarding the characteristics of human visual memory in the spatial-board task.

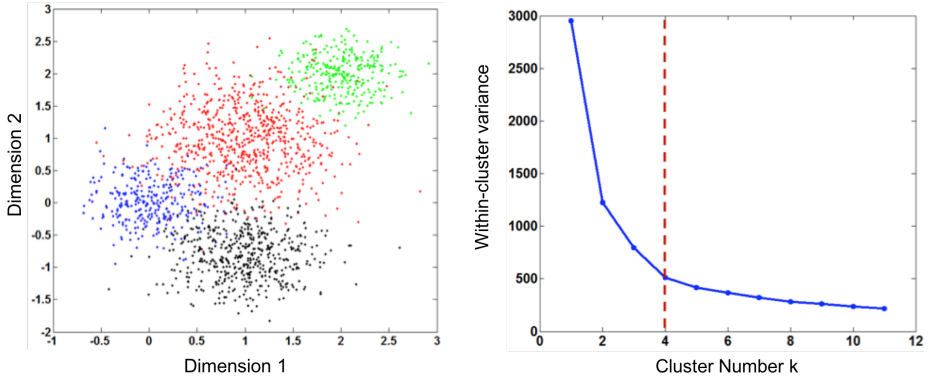
4.1 Algorithm: Determination of Cluster Number

A straightforward idea to study how many kinds of visual memory difficulty profiles exist is to perform clustering on pre-collected data. The number of clusters will tell us the number of different visual memory difficulty profiles. However, a key challenge here is to determine the number of clusters without a prior knowledge about the data. In fact, many clustering algorithms suffer from the limitation that the number of clusters has to be specified by a human user [117–119]. Consequently, there have been a number of approaches published in the literature for choosing the right k after multiple runs of k-Means [120–122], being a very popular machine learning clustering algorithm. The notion of a *cluster* is not uniquely-defined as

it depends on the form of the evaluation function. In order to find the appropriate number of clusters, some approaches [123–125] construct an evaluation graph by taking the x-axis as the cluster number and the y-axis as the corresponding evaluation function value. One can then examine the characteristics of such an evaluation graph to determine the number of clusters. A basic idea is to identify the *knee/ elbow* of the evaluation graph. Figure 4.2 (b) and (d) show the evaluation graphs for the two datasets with 3 Gaussian clusters (see Figure 4.2 (a)) and 4 Gaussian clusters (see Figure 4.2 (c)) respectively. In these graphs, the within-cluster variance is used as the evaluation metric. We can see the evaluation graph is monotonically decreasing as the within-cluster variance will decline as the cluster number k increases. However, the decrease in the within-cluster variance would become much smaller when k surpasses the true cluster number, as after this point creating more clusters only lead to partitions within groups rather than between groups [124]. Therefore, one can visually inspect the *knee* of the evaluation curve which corresponds to the correct number of clusters, as shown in Figure 4.2.



(a) Gaussian cluster with 3 components (b) Evaluation graph



(c) Gaussian cluster with 4 components (d) Evaluation graph

FIGURE 4.2: Visual inspection of the *knee* in the evaluation graph.

However, determining the *knee* position of the evaluation curve is actually a non-trivial problem. The visual inspection method is ambiguous especially when there is a high degree of intermix between the clusters. Salvador and Chan [123] proposed to determine the *knee* by finding the pair of lines that most closely fit the curve (with the minimum total root square error) and returning the intersection of these two lines as the *knee*. But their method is mainly focused on hierarchical clustering and segmentation algorithms, of which the evaluation graph is usually non-smooth and not monotonically decreasing/increasing as is in our case. To solve this problem, this section proposes a new method to find the *knee* of the evaluation graph by analyzing and exploiting the curvature information of the evaluation graph. Works in [126] and [123] have briefly mentioned the idea of employing the maximum curvature point to identify the number of clusters, however none of them formally defined and applied the curvature-based method nor discussed the relative challenges and limitations. In this section, an in-depth discussion on how to use curvature to find the *knee* in the evaluation graph is presented.

Specifically, the contributions in this section are threefold. Firstly, the information in the curvature is exploited to find the *knee* in the evaluation graph to reduce the ambiguity inherent in the process of visual inspection. Secondly, the challenges and limitations of such curvature-based method are analyzed. Finally, to overcome these challenges to find the cluster number, a new curvature-based heuristic rule is proposed. The proposed method is evaluated on a wide range of synthetic and real-world datasets.

4.1.1 Background

The issue of determining the clustering number k is a major challenge in cluster analysis. To address this problem, numerous approaches have been suggested over the years. A popular approach is to use an evaluation graph that is constructed by plotting within-cluster variance $J(k)$ for a clustering procedure against the number of clusters k employed. Using the raw J(k) function to identify the number of clusters k is impossible since J(k) itself monotonically decreases when k increases. Nonetheless, as Sugar et al. [124] pointed out, the evaluation graph actually contains the necessary information for choosing the correct cluster number. Some previous works have analyzed the evaluation graph in the presence and absence of

the clusters and proposed more sensitive characteristics to determine the cluster number.

Some early efforts proposed heuristic indexes to determine cluster numbers. Calinski et al. [120] suggested an index with the F-test form based on within-cluster variance. The method is the best performer in the experiments conducted by Milligan and Cooper [127]. Krzanowski and Lai [122] also derived a criterion using within-cluster variance for choosing clustering number and proposed a plausible stopping rule. In their work, this criterion outperformed Marriott's approach [128], which used within-cluster determinant, rather than within-cluster variance. Another popular heuristic rule was developed by Hartigan et al. [121] based on the intuition that for $k < k^*$, where k^* is the optimal cluster number, $J(k+1)$ is drastically smaller than $J(k)$, however, for $k > k^*$, $J(k+1)$ and $J(k)$ are not that different. In the experimental study comprising 8 cluster methods [129], Chiang et al. found that the Hartigan's rule can give potentially the best performance in terms of reproducing cluster number k, however the performance deteriorates quickly when the clusters are not well separated. Besides deriving measurements based on within-cluster measure, some approaches compared the within-cluster cohesion with between-cluster separation. Kaufman and Rousseeuw [130] introduced the concept of silhouette width to measure how well each point is clustered by difference between within-cluster tightness and separation from other groups. This method demonstrated good performance in the experiment conducted by Pollard et al. [131]. Recently, an approach to the problem of estimation of the number of clusters which relies on cross-validation method was proposed by Fu and Perry [132]. The authors consider the task of choosing an optimal value of k as a model selection problem and address it via a form of Gabriel cross-validation. The value of k with the smallest prediction error is selected. The experiments show that the proposed method has competitive performance, especially in high-dimensional settings with heterogeneous or heavy-tailed noise. In addition, there are some recent studies utilizing model-based measures to examine the *knee* phenomenon. Tibshirani et al. [125] proposed a statistical procedure (gap statistic) to formalize the heuristic process of finding the location of the *knee* on the evaluation graph. The idea is to compare the evaluation graph with its expectation under an appropriate null reference distribution of the data. The method is widely used in the bioinformatics community. Unfortunately, it requires heavy computation and may even fail for larger datasets because of the matrix computing problem. Based on the Gaussian

distribution model, Sugar et al. [124] introduced the ideas from the field of rate distortion theory to examine the graph's functional form in both the presence and absence of clustering. From their mathematical derivation and empirical studies, the graph, when transformed to an appropriate negative power, can exhibit a sharp jump at the optimal cluster number. This method is computationally efficient but picking an appropriate transformation power is a non-trivial problem. Besides exploring the property of evaluation graph which is focused on post-processing the results of clustering algorithms to determine the number of clusters, there are some studies on the clustering methods in which the number of clusters can be automatically founded. A recent method proposed by Rodriguez and Laio [133] is a density-based clustering approach. Cluster centers are identified as points with higher densities than their neighbors and by relatively large distances from the points with higher densities, and the number of clusters arises intuitively after the cluster centers are determined. Tasdemir et al. [134] proposed an automated clustering method for self-organizing maps (SOMs). In this method the number of clusters is determined either by using various cluster validity indices or by prior knowledge on the considered dataset.

4.1.2 Algorithm

4.1.2.1 Maximum Curvature Point

For the simplicity of discussion, the cluster results of k-Means is used to compute the within-cluster variance as the evaluation metrics to construct evaluation graph:

$$J(k) = \sum_{j=1}^{k} \sum_{x_i \in \mathcal{C}_j} ||\mathbf{x}_i - \bar{\mathbf{x}}_j||^2 \tag{4.1}$$

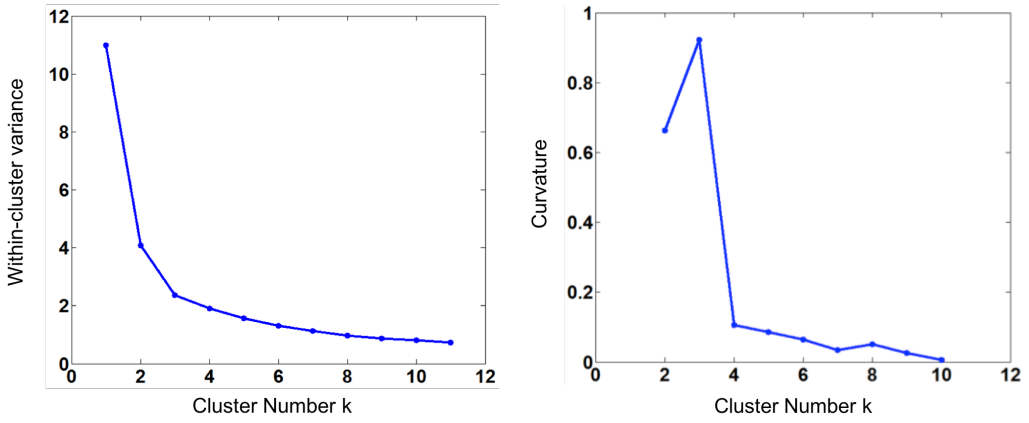
where $\mathcal{C}_j$ is the set of samples belonging to class j and $\bar{\mathbf{x}}_j$ is the sample mean of class j.

We propose to use curvature to identify the *knee* of the evaluation graph (4.1) in order to reduce the ambiguity stemming from the process of visual inspection. In mathematics, curvature is the amount by which a geometric object deviates from being flat, or straight in the case of a line. So the *knee* in the graph should

correspond to the point with the maximum curvature. For a curve explicitly given as $y = f(x)$, the curvature is defined as:

$$\kappa = \frac{|y''|}{(1 + y'^2)^{3/2}}. \tag{4.2}$$

As an example, this curvature method is applied to the real-world dataset *Seed* [1] from the University of California Irvine Machine Learning Repository [135] (UCI), which contains real application data collected in various fields and is widely used to test the performance of different machine learning algorithms [136][137]. The wheat varieties, Kama, Rosa and Canadian, characterized by measurements of main grain geometric features obtained by X-ray technique, have been analyzed. The within-cluster variance and the corresponding curvature graph are presented in Figures 4.3(a) and 4.3(b), respectively. As shown in Figure 4.3(b) the true cluster number (equal to 3) in fact corresponds to the maximum curvature point.



(a) Evaluation graph (b) Curvature of the evaluation graph

FIGURE 4.3: Dataset *Seed* [1] with real class number equal to 3: (a) Scaled cost function of k-Means; (b) Curvature of the scaled cost function.

While computing the curvature, a critical problem was discovered. It was observed that the position of the maximum curvature point changes when the original data is rescaled. In particular, when the original data is rescaled by a as shown in Equation 4.3,

$$\mathbf{x}_i = a\mathbf{x}_i, \qquad a \in \mathbb{R}^+ \tag{4.3}$$

the within-cluster variance exhibits a linear change as indicated in Equation 4.4:

$$\begin{aligned} J_a(k) =& \sum_{c=1}^{k} \sum_{x_i \in \mathcal{C}_j} ||a\mathbf{x}_i - a\bar{\mathbf{x}}_j||^2 \\ =& a^2 \sum_{c=1}^{k} \sum_{x_i \in \mathcal{C}_j} ||\mathbf{x}_i - \bar{\mathbf{x}}_j||^2 \\ =& a^2 J(k). \end{aligned} \tag{4.4}$$

Curvature is a kind of geometric property of the graph and tightly related to the range of the two axes. In the evaluation graph, the x-axis is the number of clusters, the difference of which is always one and the y-axis is the within-group variance, the range of which lies in a large variety (often much bigger than the range of x). When the original data is rescaled, the x-axis remains the same and the y-axis has a linear change as indicated in Equation 4.4. However, when the curvature is computed from the rescaled evaluation graph as in Equation 4.5, the change in the curvature is non-linear.

$$\kappa_a(k) = \frac{|a^2 J''|}{(1 + a^4 J'^2)^{3/2}} = \beta(k)\kappa(k) \tag{4.5}$$

where

$$\beta(k) = a^2 \left(\frac{1 + a^4 J'^2}{1 + J'^2} \right)^{\frac{3}{2}}.$$

For each k, the change of curvature $\beta(k)$ is not only related to a, but also to $J^{'}$. This non-linear change in the curvature will cause the shift of the maximum curvature point. In fact, it can be easily proven that for $a > 1$ (when the data is enlarged), the maximum curvature point moves rightwards and for $a < 1$ (i.e. the data is shrank), the point with maximum curvature moves leftwards. As we know, while rescaling the data, the cluster structure in fact remains the same and so does the cluster number. Therefore, although the raw curvature can serve as an effective way to identify the *knee* of the evaluation graph, it is indeed a poor indicator of cluster number. It should be noted that the traditional *knee* method suffers from the same scaling problem. When the within-cluster variance against k is plotted, the software usually automatically scales the range of axes for representation purpose because the range of within-cluster variance is often much bigger than k (see Figure 4.2). When the *knee* of the graph is inspected visually,

it is actually being examined under some scaling factor and thus the results may be unreliable.

Our goal is to eliminate the influence of the scaling factor and at the same time still exploit the usefulness of curvature in the detection of the *knee* on a graph. To this end a new curvature-based index is proposed which does not depend on the scaling factor.

4.1.2.2 Beyond Curvature

Firstly, the impact of the scaling parameter on the change of curvature for each k was analyzed. For convenience, let's define a scale parameter $\alpha = a^2$. For each point on the graph, Figure 4.4 plots curvature κ against the scaling parameter α using two datasets (*Ionosphere* [2] and *Breast tissue* [3]) from the UCI. Examining the curve, one can observe the following properties:

- All the curves are bell-shaped lines. On each line, the peak occurs between $\alpha = 10^{-3}$ and $\alpha = 10^{3}$. The curvature approaches zero when $\alpha \ll 10^{-3}$ or $\alpha \gg 10^{3}$;
- The location of the peak depends on k, and
- Also, the peak value differs with respect to k.

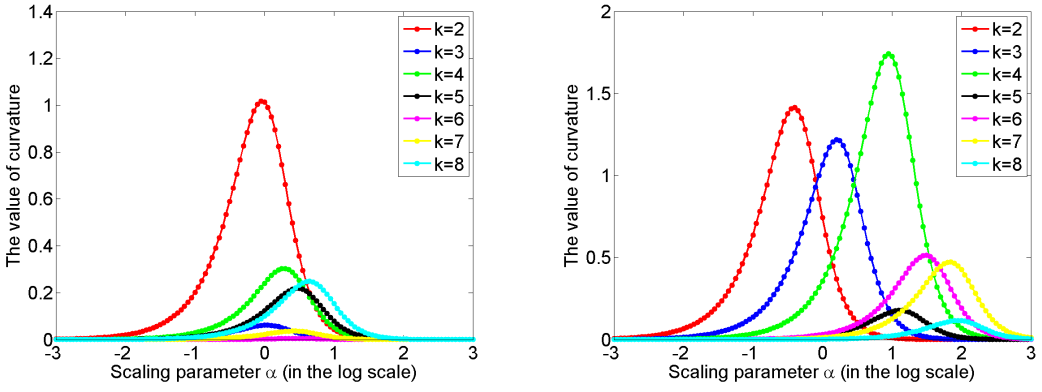


FIGURE 4.4: Plot of curvature-scale parameter for two datasets: (a) *Ionosphere* [2] (the real class number = 2); (b) *Breast tissue* [3] (the real class number = 4).

Furthermore, there is an interesting phenomenon that the k with the highest peak value corresponds to the true cluster number for the two datasets in our experiment. In *Ionosphere* dataset, which has two clusters, the peak value at $k = 2$ is the highest and for the *Breast tissue* dataset the highest peak appears at $k = 4$, which again corresponds to the true number of clusters. In the reminder of this section, this phenomenon is further investigated from the mathematical point of view.

From the analysis before, we can see the curvature is related to both cluster number k and scale parameter α as

$$\kappa(\alpha, k) = \frac{|\alpha J''(k)|}{\left(1 + \alpha^2 J'(k)^2\right)^{3/2}}. \tag{4.6}$$

Our goal is to focus on the influence of k and to eliminate the effect of α. So this work proposes to choose the optimal k by solving the following optimization problem.

$$K = \arg\max_k \max_\alpha \kappa(\alpha, k) \tag{4.7}$$

Let's start with computing $\frac{\partial \kappa}{\partial \alpha}$

$$\frac{\partial \kappa}{\partial \alpha} = \frac{|J''|}{(1 + \alpha^2 J'^2)^{\frac{5}{2}}}(1 - 2\alpha^2 J'^2), \qquad \alpha > 0. \tag{4.8}$$

From Equation 4.8 we can see that κ is a concave function with respect to α. For each cluster number k, κ reaches its maximum value if and only if $\alpha = -\frac{1}{\sqrt{2}J'}$. The maximum value is denoted as Equation 4.9:

$$\max_\alpha \kappa(\alpha, k) = \frac{1}{\sqrt{2}(\frac{3}{2})^{\frac{3}{2}}} \times |\frac{J''}{J'}|. \tag{4.9}$$

Based on Equation 4.7 and Equation 4.9, the k with the highest peak value is chosen to be returned, i.e.

$$K = \arg\max_k |\frac{J''(k)}{J'(k)}|. \tag{4.10}$$

Now, let us explore the meaning of Equation 4.10 based on finite difference approximations of first order and second order derivatives. Let's define

$$\det_k = J(k-1) - J(k),$$

that describes the decrease of within-cluster variance (i.e. the increase of between-cluster variance) from $k-1$ clusters to k clusters. Since within-cluster variance $J(k)$ is monotonously decreasing, we have

$$\det_k \geq 0, k = 2, 3...$$

For each cluster number k, Equation 4.10 can be rewritten as:

$$\arg\max_k \frac{J''(k)}{J'(k)} = \arg\max_k |\frac{\det_k - \det_{k+1}}{\det_{k+1}}|. \tag{4.11}$$

Therefore, the Curvature method relies on the ratio of two consecutive decreasing amounts for each k. This method is in favor of bigger $\det_k$ and smaller $\det_{k+1}$ such that a decrease of within-cluster variance from $k-1$ to k is relatively large while from k to $k+1$ is relatively small. This conforms to the *knee* method, which is based on the idea that one should choose a number of clusters so that adding another cluster would not provide much better modeling of the data. The proposed method is summarized in Algorithm 1.

Algorithm 1: Algorithm for the Curvature method

For $k = 1 : (k_{\max} + 1)$
  For $t = 1 : 20$
    Run the k-Means algorithm to compute the within-cluster variance:
    $$j(k,t) = k\text{-Means}(k)$$
  Take the minimum within-cluster variance across multiple times:
  $$J(k) = \min_t j(k,t)$$
For $k = 2 : k_{\max}$
  Compute the Curvature index : $r(k) = |\frac{J''(k)}{J'(k)}|$
Return the optimal number of clusters K with the maximum value of $r(k)$:
$$K = \arg\max_k r(k)$$

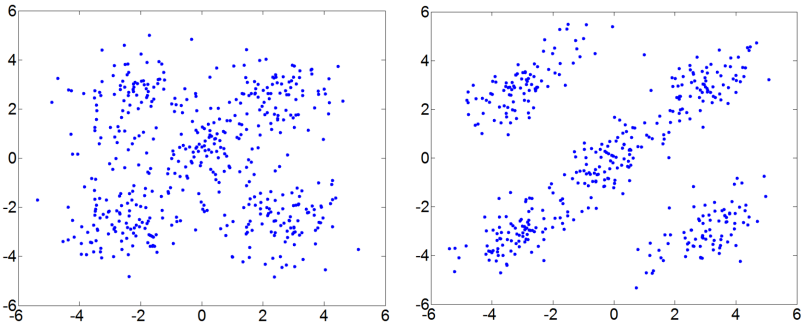
4.1.3 Experimental Results

The proposed method was compared with 6 other well-known approaches of comparable computational complexity: the CH method [120], the KL method [122], the Hartigan method [121], the Silhouette method [130], the Gap method [125], the

Jump method [124]. The detail descriptions of these six approaches can be found in Appendix B.1.

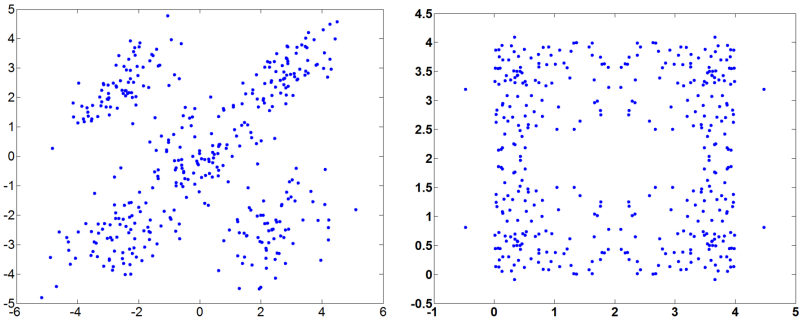
4.1.3.1 Experimental Results on Synthetic Datasets

In this section the performance of the proposed Curvature method is investigated from three aspects using synthetic data. Firstly, the experiment compares the accuracy of estimating the optimal number of clusters with our method with the other six chosen approaches. Secondly, the ability of the Curvature method to identify hierarchical cluster structure is examined. Lastly, the performance of Curvature index is evaluated with different extents of intermix/overlap between clusters.



(a) Simulation 1: five Gaussian clusters.

(b) Simulation 2: five Gaussian clusters with a correlation of 0.7 among dimensions.



(c) Simulation 3: five Gaussian clusters with differing correlation among dimensions.

(d) Simulation 5: four exponential clusters.

FIGURE 4.5: Data generated in simulations 1, 2, 3 and 5. (Simulation 4 is in high dimension and is not shown here.)

Firstly, regarding the evaluation of the accuracy of cluster number, some works in the literature have discussed data generation issues for experimental comparison

of various methods. Here we borrowed the basic ideas from [124, 129] and designed the five experiment settings considering the factors of within-cluster spread, between-cluster separation, the number of dimensions, the dependence among dimensions and Gaussian/non Gaussian distribution structure (see Figure 4.5). The experiment settings details are shown in Appendix B.2.

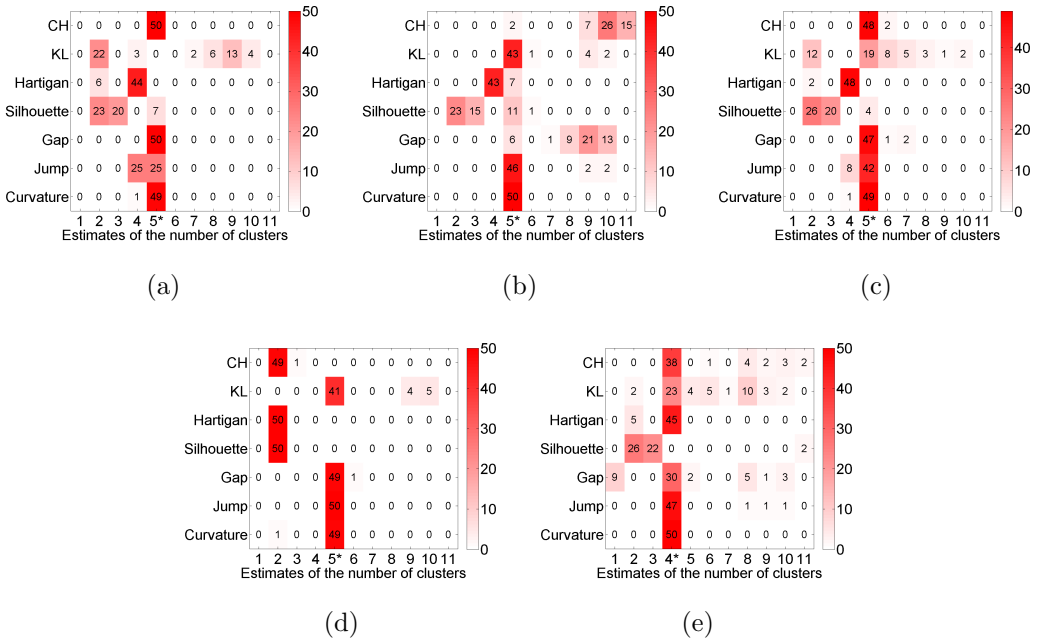


FIGURE 4.6: Results for synthetic data illustrated using heat maps. The numbers represent the respective counts of 50 trials for each method in each simulation. The true number of clusters are highlighted with star (*) symbols.

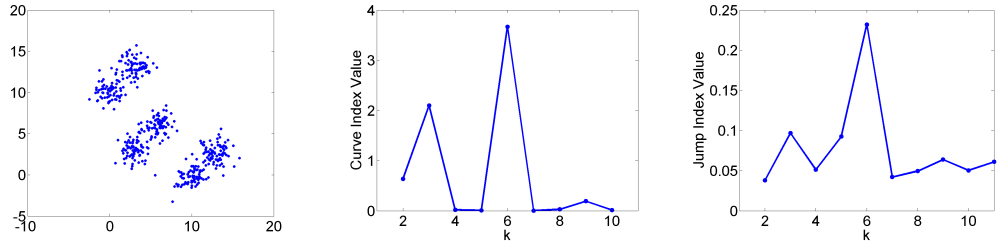
TABLE 4.1: Average running time of the seven methods (in seconds).

CH	KL	Hartigan	Silhouette	Gap	Jump	Curvature
0.0006	0.0005	0.0006	1.794	11.92	0.0006	0.0005

The results are shown in Figure 4.6. The proposed method (Curvature) achieved at least 98% accuracy in each of the 5 simulations, which was visibly the highest performance score. Among the comparative approaches, the best score was accomplished by the Jump method (Jump), followed by Gap algorithm (Gap). Moreover, as shown in Table 4.1, the proposed method achieved the lowest running time of about 0.5 ms.

Next, the reasonability of the proposed index is further analyzed based on its ability to detect hierarchical cluster structure. In this experiment the proposed method is

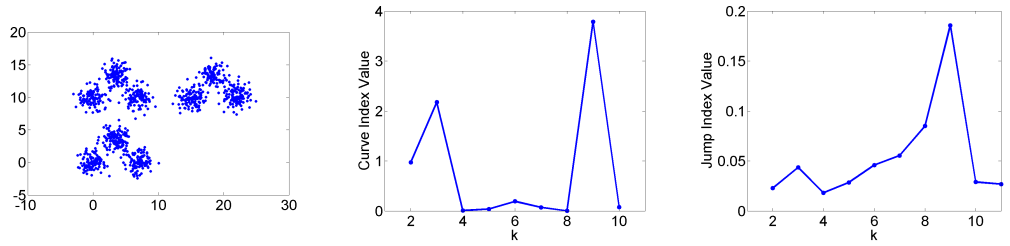
compared with the Jump algorithm since among 6 approaches the ability to identify hierarchical cluster structure was reported only for that method. A simulation with hierarchical cluster structure is designed (see Figure 4.7 (a)), which contains a two-dimensional mixture of six Gaussian clusters evenly spaced in 3 distinctive groups and each group consists of two components. The results of the Curvature method and Jump method are shown in Figures 4.7 (b) and (c) respectively. Both methods returned number 6 as the optimal candidate for the number of clusters and presented two peaks at k=6 and k=3. Therefore, both of them demonstrated the *in principle* ability to detect hierarchical cluster structure. However, in the proposed method the two peaks (corresponding to k=6 and k=3) are very distinctive and a deep valley appears between them (see Figure 4.7(b)). The values assigned to other selections of k are penalized as being very small. In the Jump method, the two peaks are less distinctive (see Figure 4.7(c)). In fact, the peak for k=3 is too small to provide any reliable information about hierarchical cluster structure in the practical application. And the value for k=5 is even a little higher than that for k=3 and is wrongly returned as the second candidate.



(a) 6 clusters in 3 groups (b) Curvature method (c) Jump method

FIGURE 4.7: Performance of hierarchical cluster structure detection with 6 Gaussian clusters spaced in 3 groups.

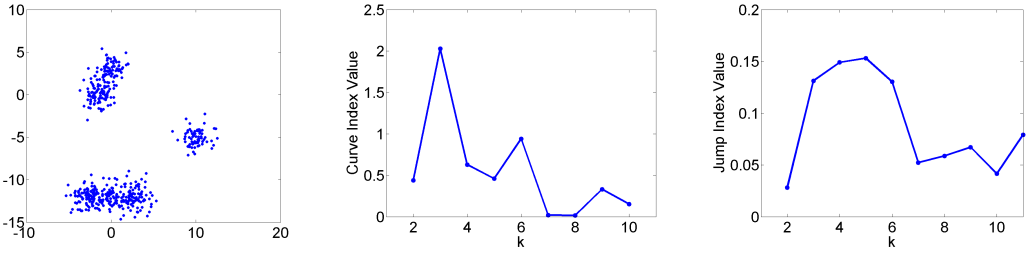
Another two examples of hierarchical cluster structure detection are given in Figure 4.8 and Figure 4.9. In Figure 4.8, the simulation consists of 9 clusters spaced in 3 identical groups (see Figure 4.8(a)). Each group contains 3 clusters. In the result of Curvature method, there are distinctive peaks at k=3 and k=9 (see Figure 4.8(b)). Nevertheless, in the Jump approach, the values for $k = 6, 7, 8$ are all greater than the value at k=3, which means the detection of hierarchical cluster structure is unsuccessful (see Figure 4.8(c)). Similarly, Figure 4.9 shows the results for 6 Gaussian clusters spaced in 3 groups which contain 1, 2, 3 clusters respectively. Again, the Curvature approach identifies a large peak at k=3 and a small



(a) 9 clusters in 3 groups (b) Curvature method (c) Jump method

FIGURE 4.8: Performance of hierarchical cluster structure detection with 9 Gaussian clusters spaced in 3 groups. Each group consists of two components spaced in a line with a separation of 2.

peak at k=6. However, the Jump approach fails to detect the correct number of clusters in this simulation. The experiment results for these two examples provide evidence that our approach is more effective in cluster structure detection than the Jump method. All in all, the results suggest that some useful information can be obtained from the index graph in the proposed method, as the distinctive peaks and the second maximum point can be used as an effective hint for the existence of hierarchical clusters. We believe this ability is an additional support for the cluster descriptive characteristics of the Curvature index.

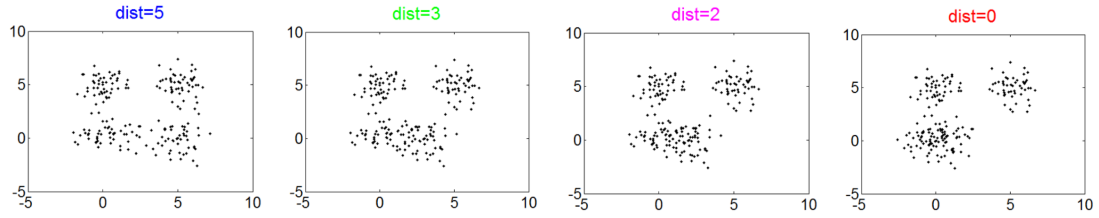


(a) 6 clusters in 3 groups (b) Curvature method (c) Jump method

FIGURE 4.9: Performance of hierarchical cluster structure detection with 6 Gaussian clusters spaced in 3 groups with means values of 6 clusters are (0, 3), (-1.5, 0), (-3, -12), (0, -12), (3, -12), (10, -5).

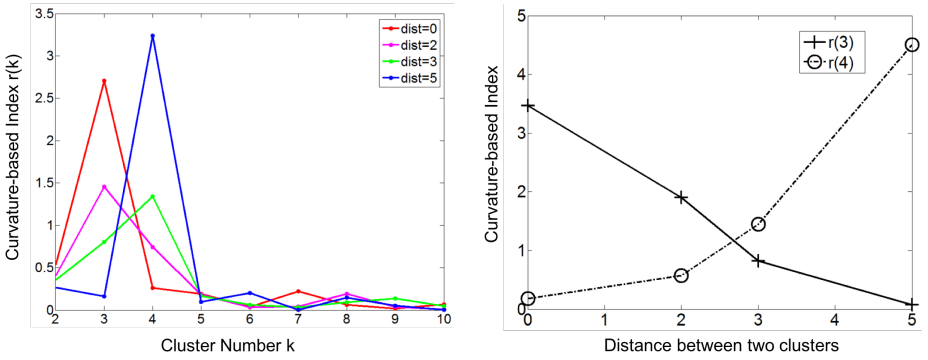
The last simulation is designed to investigate how the Curvature index value changes when the extent of intermix between the clusters varies. To this end, four Gaussian clusters in two dimensions which are spaced in a square with the side length of 5 are generated. Each cluster has a standard deviation of 1 in each dimension. Then the intermix between clusters is introduced by moving one cluster closer to another. More specifically, the distance between two clusters is varied

from 5 to 0. Figure 4.10 presents the datasets with distances equal to 5, 3, 2 and 0. It is clear from the figure that the initial dataset contains 4 clusters and the last one contains 3 clusters. The second and the third ones record the transition from 4 to 3 clusters. The Curvature index graphs are shown in Figure 4.11. The Curvature method estimates the first two datasets as consisting of 4 clusters and the other two as consisting of 3 clusters. Since the definition of a cluster is not precise,



(a) Dataset #1 (b) Dataset #2 (c) Dataset #3 (d) Dataset #4

FIGURE 4.10: Simulated compounded datasets # 1-4 with distances between two clusters at 5, 3, 2, 0 respectively.

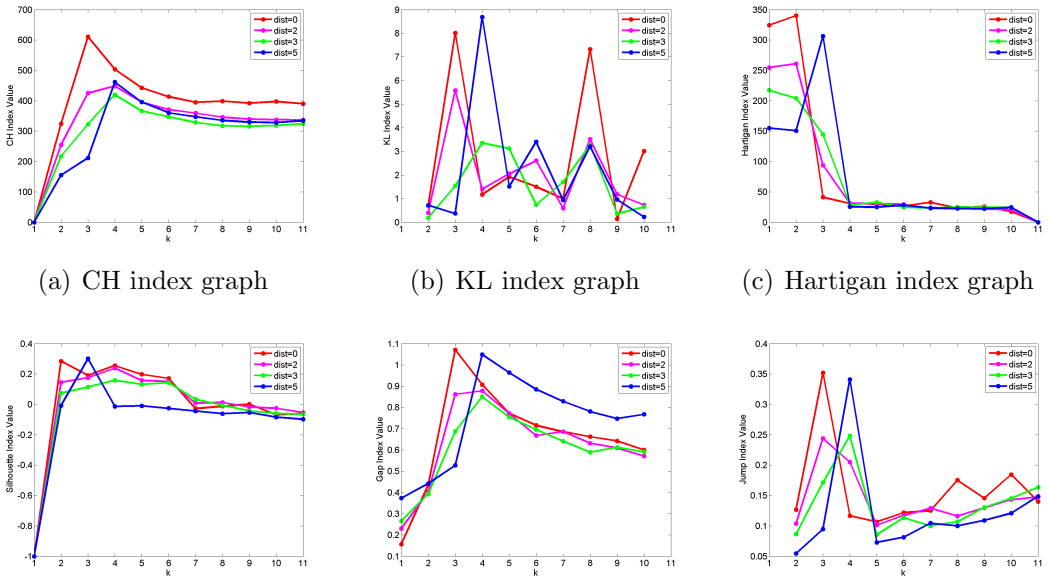


(a) The Curvature index graph for four datasets (b) The Curvature index at k=3 and k=4 for four datasets

FIGURE 4.11: Curvature index graphs for compounded data.

we choose not to judge whether the results are correct or not. Instead we seek to analyze the reasonableness of the Curvature index. Firstly, it can be observed from Figure 4.11(a) that for all the four datasets the maximum point appears either at $k = 3$ or $k = 4$ and the values for other k are very low, which indicates the low noise level of the proposed Curvature index. Secondly, as the distance between the two clusters increases, the Curvature index at $k = 4$ decreases and Curvature index at $k = 3$ increases. This phenomenon corresponds very well to the fact that the data is actually transformed from 4 to 3 clusters. Another interesting observation is that the maximum peaks in the index graph of datasets #1 and #4 are more

distinctive than the peaks in datasets #2 and #3. This conforms to the fact that datasets #1 and #4 have clearer cluster structure than the other two datasets. The performance of the 6 existing approaches on the four datasets are shown in Figure 4.12. It can be seen that the maximum points generally appear either at $k = 3$ or $k = 4$ in the evaluation graphs of CH, KL, Hartigan, Gap and Jump methods. However, in the case of CH, Hartigan and Gap plots, these peaks are not obvious (see Figure 4.12(a)(c)(e)). Similar to the Curvature method, KL and Jump plots also have distinctive peaks at $k = 3$ and $k = 4$, but these two methods have much higher noise level than the Curvature method, because the index values are not properly penalized in these approaches for $k > 4$ (see Figure 4.12(b)(f) and Figure 4.11(a)). Based on the above results and analysis, we conclude that the Curvature index behaves reasonably and proportionally to increasing intermixing among the clusters.



(a) CH index graph (b) KL index graph (c) Hartigan index graph

(d) Silhouette index graph (e) Gap index graph (f) Jump index graph

FIGURE 4.12: Performance of the 6 comparative approaches on four datasets presented in Figure 4.10.

4.1.3.2 Experimental Results on Real-World Datasets

In order to verify the efficacy of the proposed method on non-artificially constructed datasets, the Curvature method is further compared with six other approaches on 20 real-world datasets from the UCI. The cumulative accuracy in estimating the

number of clusters for all tested datasets is given in Table 4.2. (More detailed results of the experiment are presented in Appendix B.3.)

TABLE 4.2: Performance of seven approaches on 20 real world datasets.

	CH	KL	Hartigan	Silhouette	Gap	Jump	Curvature
First candidate accuracy	40%	40%	0	35%	15%	15%	50%
First two candidates accuracy	50%	60%	0	55%	25%	15%	80%

The proposed method also appears to be the most robust among the tested algorithms when applied to the real-world datasets. It achieved the highest accuracy and was able to produce correct results for 10 out of 20 sets. As the real-world data poses significant challenges to generally all methods, we further present the comparative performance for the correct estimation of the true number of clusters in one of the first two candidates (top-2 selection). In this comparison, the Curvature method achieved 80% accuracy, while the highest accuracy obtained among the other methods was 55%, with the Silhouette method.

In a more detailed comparison of both methods (the Silhouette and the Curvature), it is observed that the Silhouette method seems to favor small cluster numbers. In particular, as presented in Table 4.3, for $k < 4$ its top-2 selection accuracy equals 73.3%, while for $k \geq 4$ the top-2 selection accuracy equals zero. In comparison, the top-2 selection accuracies of the Curvature method were 86.7% and 60%, respectively, suggesting higher robustness and lesser sensitivity to cluster count.

TABLE 4.3: Top-2 selection accuracy of seven approaches on 20 real world datasets.

	CH	KL	Hartigan	Silhouette	Gap	Jump	Curvature
Top-2 selection accuracy($k < 4$)	67%	53%	0%	73%	13%	20%	87%
Top-2 selection accuracy($k \geq 4$)	0%	40%	0%	0%	40%	0%	60%

When it comes to the datasets with a large size, such as *MiniBooNE* [138] with $130,064$ instances in 50 dimensions or *Skin* [139] with $245,057$ instances in 3 dimensions, the Curvature Method has an advantage of computational efficiency. The Silhouette method and the Gap method are unable to handle such datasets due to their matrix computation problem.

4.1.4 Discussion

In this section, a curvature-based method to estimate the optimal number of clusters is proposed. The algorithm is computationally efficient and parameter-free. The comparative evaluation on both synthetic and real-world datasets shows that the proposed Curvature method outperforms the six other cluster count estimation algorithms. In addition, empirical results indicate that the proposed method is able to provide reliable information in terms of identifying the underlying hierarchical structure of the data. Empirical observations suggest that the Curvature method is more suitable for datasets with cluster counts smaller than 10. Beyond that limit, the cluster number yielded by the method is likely to be biased towards a smaller value. Another limitation of the proposed method is that the Curvature index value is undefined for null distributions (i.e. for the case of one cluster in the dataset). One possible remedy is to introduce an additional artificial cluster located far away from the original data. In that case, if the Curvature method returns 2 as the optimal cluster number, the original data can be regarded as coming from a null distribution. Theoretically, the proposed method can work with virtually any clustering method. However, the within-cluster curve may differ slightly for different clustering methods. This work focused the experimental results on the highly-popular k-Means algorithm. Investigation of the suitability of the proposed Curvature method for other clustering methods is a subject for future research.

4.2 Application in Visual Memory Game for Difficulty Ranking Personalization

4.2.1 Clustering-based Difficulty Ranking Personalization

The previous section proposed a method for the determination of cluster number in K-means clustering. This section describes how this method can be applied to solve the difficulty ranking personalization problem of participants in the online visual memory game platform. Figure 4.13 shows an overview of this method, which consists of two steps: offline clustering and online classification.

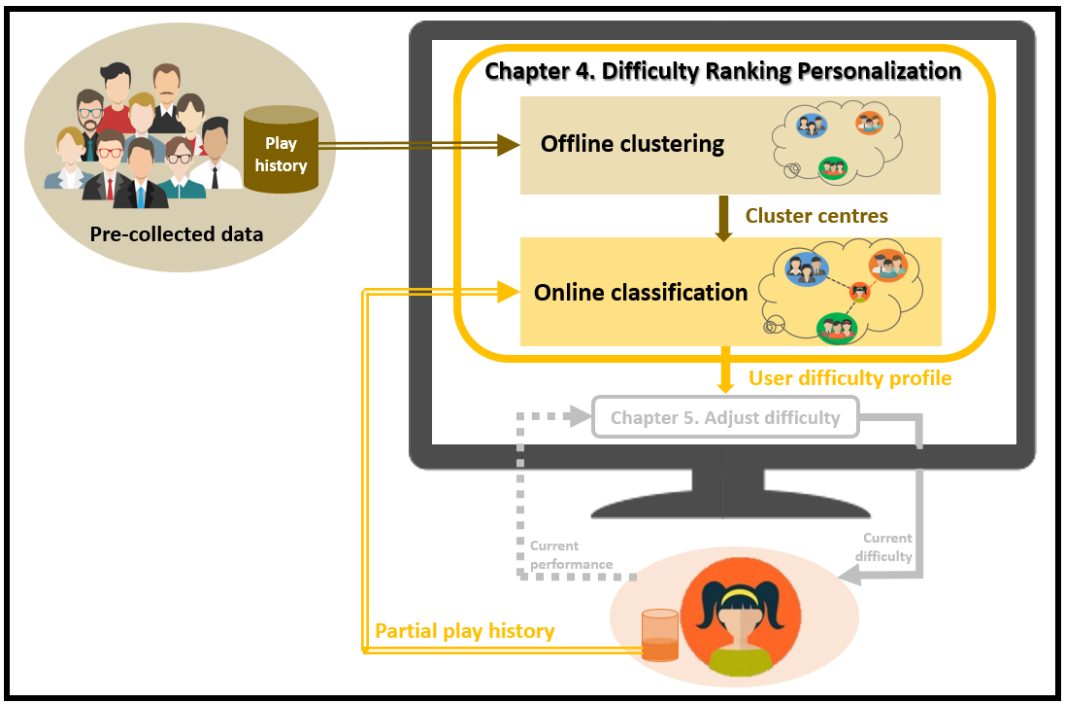


FIGURE 4.13: Overview structure of the clustering-based difficulty ranking personalization.

In the offline clustering stage, we seek to determine what kinds of difficulty ranking profiles exists in the pre-collected data by clustering. Following the notations used in Chapter 3, consider a set of tasks $a_i \in \mathcal{A}$ and a set of users $p_j \in \mathcal{P}$. The difficulty level of a task a_i for a user p_j is denoted by a scalar $D_j(a_i) \in \mathbb{R}$. During the clustering, each data sample denotes the difficulty ranking DR_j profile of a user, which includes the difficulty levels of all the tasks for the user $DR_j = [D_j(a_1), D_j(a_2), ..., D_j(a_A)]$. In other words, the user's *difficulty ranking profile* DR_j is described by a vector in the dimension of number of tasks, i.e. $DR_j \in \mathbb{R}^A$, where each dimensional component stands for the difficulty level for that specific question. In the visual memory game, there are 100 tasks in the question bank. The difficulty ranking profile data is thus in 100 dimension. By clustering the difficulty ranking profile data, we can obtain the information on how many different types of difficulty ranking profiles there are and which type of difficulty ranking profile a player should belong to. Regarding how to obtain the difficulty ranking profile DR_j for a certain user in the training data, the Pals visual memory game platform was used to collect the necessary data. Specifically, the memorization time of a task was used to measure the difficulty level of a task for this user, as stated in Chapter 3. The longer the memorization time is, the harder the task is for the

users. However, each game session has only 25 questions. This means we only know the difficulty levels for the 25 questions that were presented to the user. Therefore there is missing information in the clustering data with the values in some of the dimensions unknown. To handle this problem, a data-preprocessing step was employed. Since we only care about the relative difficulty levels instead of the absolute difficulty values, the raw difficulty levels data (i.e. the memorization time of the presented questions) was centered. Then the difficulty levels of the questions that have yet to be presented were set to be zero. After preprocessing, the proposed Curvature method was applied to determine the cluster number K and the K-Means clustering is performed. The cluster center was used to denote the *visual memory difficulty profile* of a group of participants that have similar difficulty rankings.

In the online classification stage, given a new user, we seek to determine the user's visual memory difficulty profile by assigning the user to the nearest cluster. As the new user interacts with the Pals online game, the server will record the information of the relative difficulty levels of the questions answered by the user based on the memorization time. This information forms a partial ranking on the question set. The distances between the user's partial ranking and the several profile candidates (i.e. cluster centers) were computed to choose the most similar one. To compute the distance between (partial) rankings, the Normalized Distance based Performance (NDPM) from Recommendation System literature is used [115, 140, 141]. Specifically, given a task set $\mathcal{A}$ and two difficulty ranking profiles of the question set DR_1 and DR_2, the distance $d_{DR_1,DR_2}(a_i, a_j)$ between the two ranking profiles on a question pair $a_i, a_j \in \mathcal{A}$ is defined by the following rule: if DR_1, DR_2 both agree a_i is easier (or harder) than a_j, then $d_{DR_1,DR_2}(a_i, a_j) = 0$; if DR_1 regards a_i is easier (or harder) than a_j but DR_2 regards a_i is harder (or easier) than a_j, then $d_{DR_1,DR_2}(a_i, a_j) = 2$; if DR_1 does not specify whether a_i is harder than a_j, then $d_{DR_1,DR_2}(a_i, a_j) = 0$; if DR_1 specifies whether a_i is harder than a_j but DR_2 does not, then $d_{DR_1,DR_2}(a_i, a_j) = 1$. The total distance between two given rankings is the summation of the distances on all the question pairs: $D(\mathcal{A}, DR_1, DR_2) = \sum_{(a_i,a_j)} d_{DR_1,DR_2}(a_i, a_j)$. And NDPM is the normalized version of total distance.

$$NDPM(\mathcal{A}, DR_1, DR_2) = \frac{D(\mathcal{A}, DR_1, DR_2)}{\arg\max_{DR} D(\mathcal{A}, DR_1, DR)} \tag{4.12}$$

Based on the definition of NDPM, the more the two rankings agree on which questions are harder than which, the smaller the distance is. When the two rankings fully agree on the relative difficulties of the tasks, the NDPM distance is zero and when the two rankings totally disagree, the NDPM distance is one. Furthermore, the unspecified relative difficulties in the reference ranking (DR_1) will not affect the results. Therefore, it can deal with partial ranking. During the online testing stage, as the game continues, a partial ranking is computed from the user's play history $h(t)$ based on the memorization times of the questions played so far. Then the NDPM is used to select the closest cluster center to this partial ranking. The selected cluster center is regarded as the current personalized difficulty ranking for this user. The whole algorithm for difficulty ranking personalization is shown in Algorithm 2.

Algorithm 2: Algorithm of clustering-based difficulty ranking personalization

Offline stage

Preprocess the performance data X:

$$DR = preprocessing(X)$$

Determine the cluster number:

$$k = curvatureMethod(DR)$$

Obtain the visual memory difficulty profile candidates (cluster centers):

$$c = kMeans(k, DR)$$

Online stage

For every 5 time steps $t\%5 == 0$:

Compute distances between play history $h(t)$ and candidates $c(i)$

$$d(i) = NDPM(\mathcal{A}, h(t), c(i)), i = 1..k$$

Choose the nearest cluster center $c(i_*)$ as the difficulty ranking

$$i_* = \arg\min_i d(i)$$

4.2.2 Experimental Settings

The experiment was conducted using real gameplay data gathered from the Pals online visual memory game platform described in Chapter 3. Four kinds of difficulty ranking were examined:

- Random ranking: the difficulty levels for 100 questions are generated randomly.

- Number-based ranking: the difficulty levels of the 100 questions are denoted as the number of targets they contain. For example, the difficulty levels for the 3-target tasks are 3 and 4-target tasks are 4.
- Fixed ranking: the difficulty level for a question is the average memorization time of this question in the training data. With this method, the ranking is also obtained based on training data but is same for all the players.
- Personalized ranking: the proposed clustering-based method in Table 2.

The participants were recruited from Amazon Mechanical Turk platform. The training data includes the play records of 544 players [2] from which the performance data is used for clustering. The Curvature method identified three distinctive clusters, which indicates there are three distinguishable visual memory difficulty profiles among the players. These three visual memory difficulty profiles will be further examined in Section 4.2.4. The gameplay data of another 77 subjects was used as the testing data.

4.2.3 Experimental Results

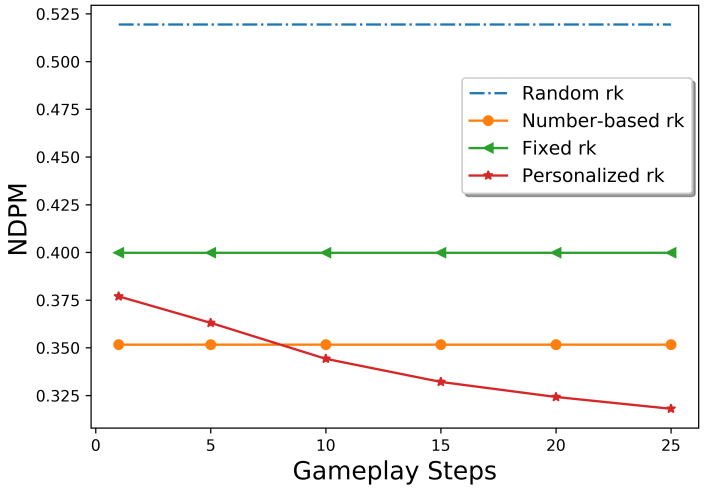


FIGURE 4.14: Difficulty ranking prediction error in terms of NDMP distance under four methods.

[2] More results with different amount of training data can be found in Appendix A.1.

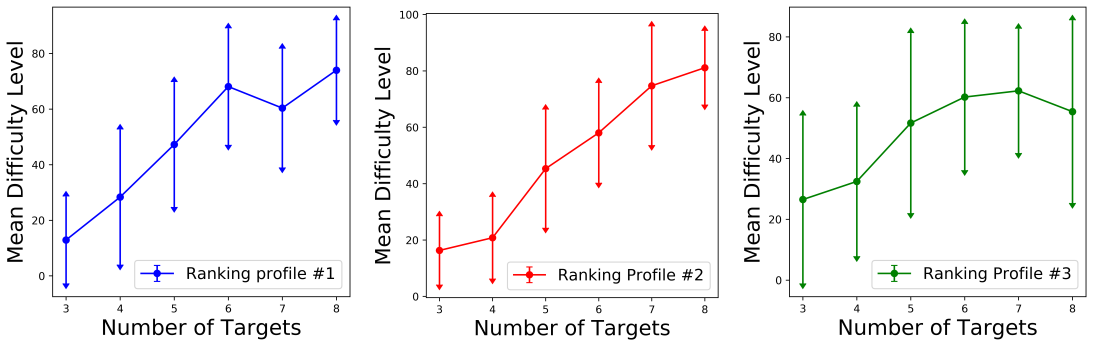
Figure 4.14 presents the average NDPM distance between the predicted ranking profiles by the four methods and the true difficulty profiles of the players. The true difficulty ranking profiles come from the memorization time in the testing data. The results show that the performance of personalized ranking method continues to improve as the game continues. The proposed DRP approach achieved the most accurate prediction, when using more than 5 steps of play history. Furthermore, we can see from the results that the clustering process used to match a player's visual memory difficulty profile to the closest of three identified visual memory difficulty profiles is actually a crucial step. The importance of clustering is further evidenced when the fixed ranking that assumes only one universal visual memory difficulty profile among users was demonstrated to perform even worst than the number-based ranking in which a simple heuristic of measuring difficulty based on target count.

4.2.4 Insights about Visual Memory

In the previous section, the clustering-based personalized difficulty ranking method was applied to the data gathered from the Pals online visual memory game and three distinctive visual memory difficulty profiles were identified. In this section, some new insights into the human visual memory are presented based on the analysis of these three visual memory difficulty profiles.

4.2.4.1 Effectiveness of Number of Targets as Difficulty Indicator of Visual Memory Task

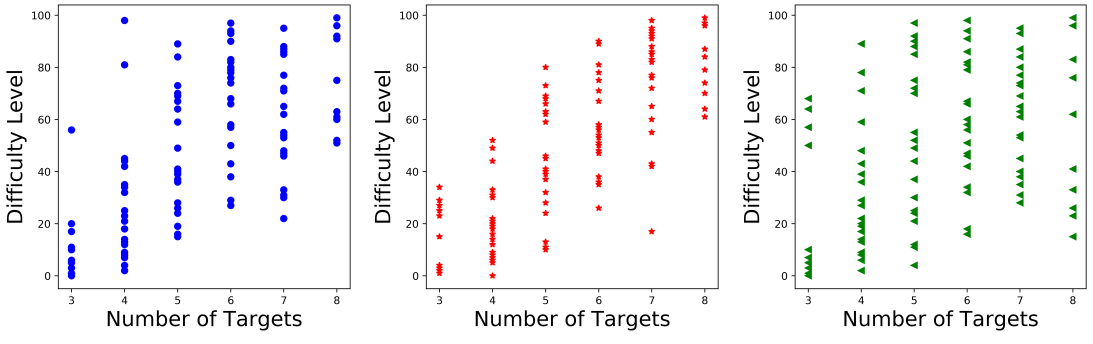
As pointed out in Chapter 3, an obvious and intuitive difficulty measure of a visual memory task is the number of the memory targets presented. Therefore, we first investigated whether the number of targets in a visual memory task can predict memorization difficulty. For convenience, the difficulty levels in the three ranking profiles were transformed to a scale of 1 to 100 using the following method. The 100 questions are sorted from the easiest to hardest based on the ranking profile. The sorted number is denoted as the difficulty level. In this way, difficulty level of 100 means the questions is the hardest one in the question bank and difficulty level of 1 means it is the easiest one.



(a) (b) (c)

FIGURE 4.15: The average difficulty levels for visual memory tasks with the same number of targets in (a) ranking profile #1; (b) ranking profile #2; and (c)ranking profile #3.

Figure 4.15 plots the average difficulty levels with respect to the number of targets. For all the three ranking profiles, their curves move generally upwards with increasing target. This indicates that increasing task difficulty correlates well with increasing target count. However, as denoted by the error bar in Figure 4.15, there are high levels of variance in the difficulty levels of the tasks with the same task number. In fact, in all three ranking profiles, the standard derivation in the difficulty levels of the same-target-number tasks are above 12 and some even reach 30. This suggests although some tasks have same amount of targets, their difficulty can vary in a wide range. To further examine the relationship between the difficult levels and the target number, the detailed predicted difficulty levels for all the questions are shown in Figure 4.16. We can see many cases where more targets



(a) (b) (c)

FIGURE 4.16: The predicted difficulty levels of 100 tasks in (a) ranking profile #1; (b) ranking profile #2; and (c)ranking profile #3.

do not necessarily make the question harder to memorize. Take the ranking profile

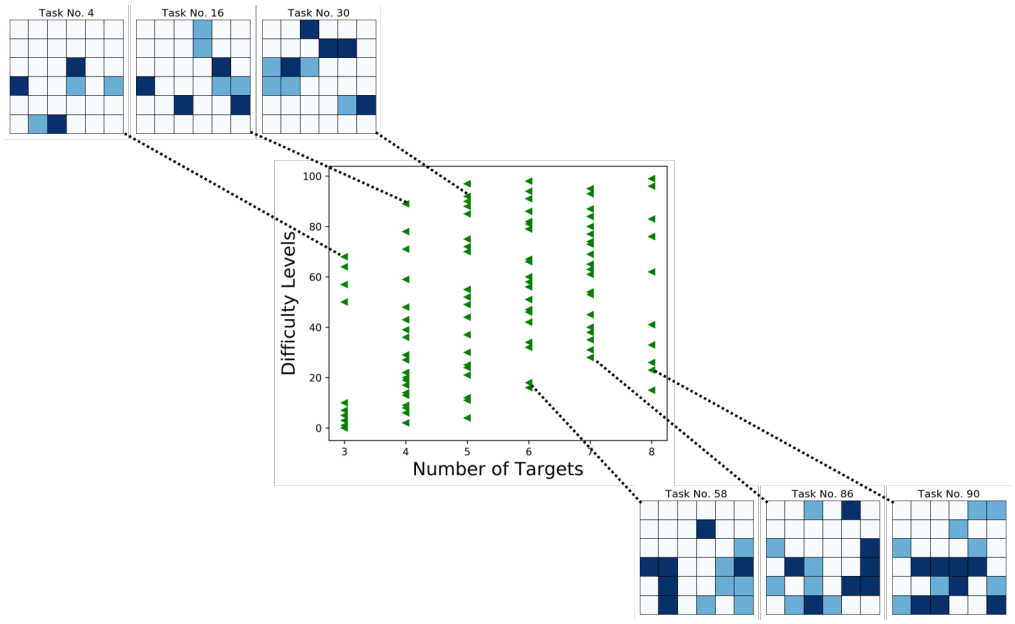


FIGURE 4.17: Difficulty level versus target number in the visual memory difficulty profile #3

#3 as an example (see Figure 4.17). In fact, although some questions like No.58, 86 and 90 contain many targets (6, 7, 8 targets) to remember, the targets actually form distinct groups with structural patterns like vertical or horizontal lines, which makes it easier to memorize. On the other hand, some questions like No.4, 16 and 30 with less but scattered targets look quite unorganized and are therefore difficult to recall.

This phenomenon was investigated from a statistical point of view. The visual memory questions were divided into three groups: small-size questions with 3 or 4 targets, medium-size questions with 5 or 6 targets, and large-size questions with 7 or 8 targets. The t-tests were conducted to detect whether one group of question is significantly harder than another according to each visual memory difficulty profile. For the ranking profile #2, the medium-size questions ($M = 51.7$, $SD = 20.4$) are significantly harder ($t(68) = 7.36$, $p < 0.001$) than the small-size questions ($M = 19.3$, $SD = 13.9$) and significantly easier ($t(78) = -5.18$, $p < 0.001$) than the large-size questions ($M = 76.8$, $SD = 19.0$). This result indicates for people with this visual memory difficulty profile, the number of targets can indeed reflect the difficulty for memorization of visual memory tasks. However, this same argument cannot be made for all our participants. Specifically, for those with the ranking profile #1, the medium-size questions ($M = 57.7$, $SD = 24.1$) are still significantly harder ($t(68) = 5.95$, $p < 0.001$) than the small-size questions ($M = 23.2$,

$SD = 23.0$) but *not* significantly easier ($t(78) = -1.28$, $p = 0.203$) than the large-size questions ($M = 64.9$, $SD = 21.3$). Similarly, for those with the ranking profile #3, the medium-size questions ($M = 55.9$, $SD = 27.3$) are still significantly harder ($t(68) = 3.90$, $p < 0.001$) than the small-size questions ($M = 30.5$, $SD = 25.8$) but *not* significantly easier ($t(78) = -0.63$, $p = 0.530$) than the large-size questions ($M = 60.0$, $SD = 24.3$). These findings suggest for some participants the target number is a good indicator of difficulty but for others it is indeed unreliable, especially when the questions contain large number of targets. One possible reason is that with more targets, there is a higher chance that the targets can come together to form recognizable visual structures (e.g. L-shape) making their memorization easier for some people.

4.2.4.2 Differences among The Three Visual Memory Difficulty Profiles

A main finding in the previous section is that the number of targets can be a good predictor of task difficulty for one visual memory difficulty profile but not another. This section further examines the differences between the three profiles. The individual visual memory differences and how that may affect the difficulty of a visual memory task is a complicated issue. Some qualitative results are presented here to provide an intuitive idea of how the human visual memory differs from person to person in terms of difficulty ranking profile. Specifically, we show several visual examples that some people find easy to remember while others find challenging. Figure 4.18 presents three tasks that are in the hardest 35% questions according to

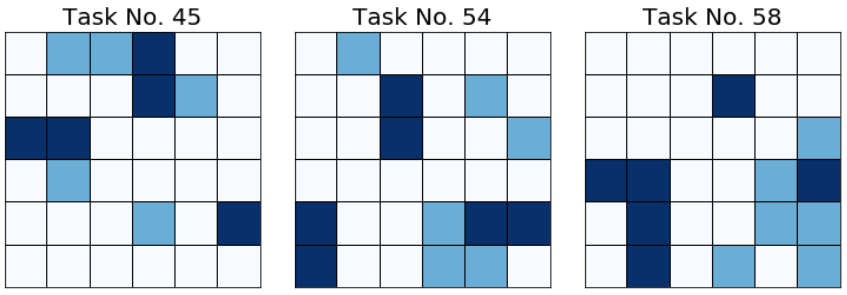


FIGURE 4.18: The question examples that ranking profile #1 finds hard and ranking profile #2 finds easy.

the ranking profile #1 (with difficulty levels as 70, 66, 94 respectively) and the easiest 35% based on the profile #2 (with difficulty levels as 32, 26, 35 respectively). On the other hand, Figure 4.19 presents some tasks that are the easiest 35% ac-

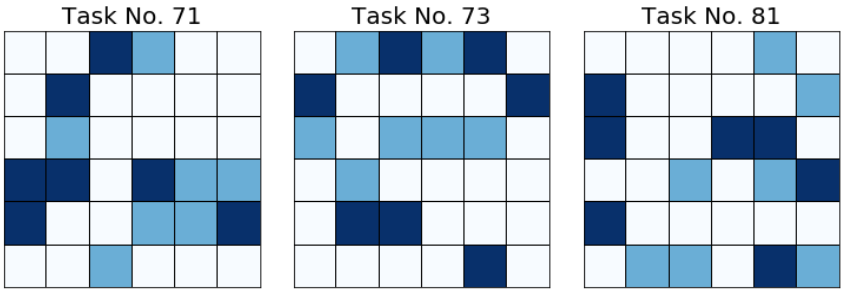


FIGURE 4.19: The question examples that ranking profile #1 finds easy and ranking profile #2 finds hard.

cording to the ranking profile #1 (with difficulty levels as 30, 22, 33 respectively) and the hardest 35% based on the profile #2 (with difficulty levels as 82, 93, 77 respectively). From these selected examples, we can get a flavor of the different visual memory characteristics that different individuals may exhibit. In the three visual memory tasks of the first example set (see Figure 4.18), the targets form visually distinct groups with vertical or horizontal line structures. Whereas those in the second example set (see Figure 4.19) are more "scattered" with no targets grouping into substantial and obvious visual structures. However, the distances between the adjacent targets seem to be closer in the second example set. It is possible that participants belonging to the ranking profile #1 are good at using relative position to memorize the target and thus find the tasks in Figure 4.19 easier. And people with the profile #2 may be better at detecting visual structures in the target layout and exploiting these while memorizing the visual pattern, which makes the tasks in Figure 4.18 easier.

TABLE 4.4: NDPM distances between the different difficulty rankings.

	NumRk	Rk#1	Rk#2	Rk#3
NumRk	0.00	0.28	0.22	0.37
Rk#1	0.20	0.00	0.32	0.43
Rk#2	0.13	0.32	0.00	0.41
Rk#3	0.28	0.43	0.41	0.00

To qualitatively measure the differences between the ranking profiles, the NDPM distances among the rankings are computed (see Table 4.4 and Figure 4.20). Interestingly, the result suggests the three personalized rankings are quite different from each other since the distances between the different personalized ranking are even greater than the distances to the number-based ranking.

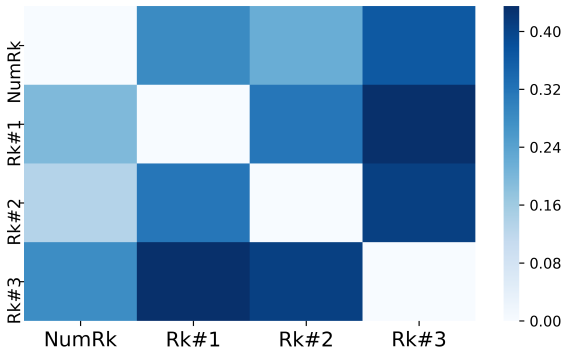


FIGURE 4.20: Heatmap of NDPM distance among difficulty rankings.

4.2.4.3 Similarities among The Three Visual Memory Difficulty Profiles

Regardless of differences in the three profiles, there are still some questions regarded as easy in all three profiles and some consistently ranked hard by most of the participants. In particular, Figure 4.21 presents the tasks that lie in the easiest 20% for all three profiles. And Figure 4.22 shows the tasks that all profiles agree that they are hard (among the hardest 30% ones). We can see the consistently

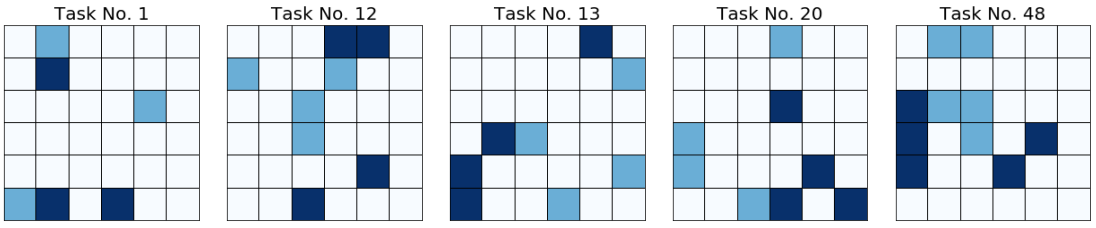


FIGURE 4.21: The five question examples that all three ranking profiles find easy (among the easiest 20%).

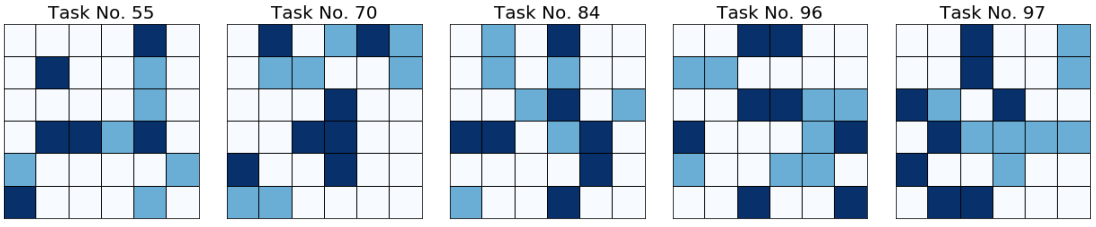


FIGURE 4.22: The five question examples that all three ranking profiles find hard (among the hardest 30%).

easy tasks are mostly small-size questions consisting of 3 or 4 targets, although

there is also a medium-size question of 5 targets (Task No.48). On the contrary, the consistently hard tasks mostly come from large-size questions consisting of 7 or 8 targets. But there is also a medium-size question of 6 targets (Task No.55). This result also corroborates the complex relationship between task difficulty and target number. To quantitatively study the similarities of the three profiles, the repetition percentage among the three profiles are computed. In particular, we examine how many questions appear in the hardest 20 questions of all three profiles and how many questions appear in the easiest 20 questions of all three rankings. Interestingly, for the hardest 20 questions, there is *no repetition* at all for the three profiles. For the easiest 20 questions, however, there are 5 questions included in all three profiles. More results concerning a varying percentage of tasks are shown in Table 4.5. We can see that the agreement of easy tasks is consistently higher than the agreement of hard tasks regardless of what percentage of tasks are taken into account. These results suggest that people seem to differ in which visual memory tasks are very hard, but have more in common when it comes to which visual memory tasks are very easy.

TABLE 4.5: Agreement among the three difficulty ranking profiles. The first row of the table is interpreted as: for the 10 hardest tasks, there are zero tasks appearing in all three rankings and for the 10 easiest tasks, there are 2 tasks appearing in all three rankings.

Number of tasks	Agreement in hard tasks	Agreement in easy tasks
10	0	2 (20.0%)
20	0	5 (25.0%)
30	5 (16.7%)	11 (36.7%)
40	13 (32.5%)	15 (45.0%)
50	24 (48.0%)	24 (48.0%)

4.2.5 Discussion

Based on the Curvature method, this section applied a clustering-based approach to obtain personalized difficulty ranking from memorization duration gathered from an online visual memory game. The personalized difficulty ranking achieved higher prediction accuracy than the traditional difficulty rankings.

The analysis of the personalized difficulty ranking led to some new insights into some of the characteristics of human visual memorization abilities. First, the quantitative evidence revealed the limitations of using the target count as the difficulty indicator. It was observed that questions containing the same number of targets do not necessarily have the same level of difficulty. High variances in difficulty levels were observed among questions with the same amount of targets. In addition, questions containing more targets does not necessarily make them harder to memorize. In some cases, the targets can form a distinctive structure which reduces the memorization difficulty. As a result, the 7-target or 8-target questions were found to be not significantly harder than 5-target or 6-target questions. Second, difficulty ranking can actually differ from person to person. The same visual task can be considered very challenging for some people but very easy for others. It seems that the users' abilities to perceive visual structure and their adoption of these target grouping memorization strategies can contribute significantly to the variability in visual memory difficulty profiles within the user population. Lastly, despite these differences between users' visual memory difficulty profiles, there are still some similarities among them. Some visual patterns are consistently considered to be easy or hard by all three difficulty ranking profiles. Interestingly, more commonality in the participants was observed for easy tasks than for the hard ones.

It should be noted that although the Curvature method predicted three visual memory difficulty profiles in the available experimental data, the ground-truth of the actual number of the distinctive profiles in human visual memory is not available. But an extensive evaluation of Curvature method has been shown in Section 4.1.3 using many other datasets with ground truth available. The result shows that the Curvature method is the most reliable approach in complex environments with high dimensions, hierarchical structure or intermix clusters.

4.3 Application in education systems

4.3.1 Experiment Settings

The proposed difficulty ranking prediction method was also applied to the real-world dataset presented in KDD Cup 2010: Educational data mining challenge [142] [3]. Three kinds of difficulty ranking were examined:

- Random ranking: the difficulty levels are generated randomly.
- Fixed ranking: the difficulty level for a question is the average response time of this question in the training data. With this method, the ranking is also obtained based on training data but is assumed same for all the players.
- Personalized ranking: the proposed clustering-based method in Table 2.

The number of question candidates is 646. Around 80% of data was used for training and the remaining data was used for testing. Specifically, in the training data, there are 385 players and 12,329 play records. In the test data, there are 49 players and 3,083 play records.

4.3.2 Experiment Results

Figure 4.23 presents the average NDPM distance between the predicted ranking profiles and the true difficulty profiles of the players. The true difficulty ranking profiles come from the response time in the testing data. The results show that the performance of the personalized ranking method continues to improve as the interaction continues. The proposed DRP approach achieved the most accurate prediction when using more than 10 steps of play history.

[3] The data is preprocessed to exclude the questions that were only played once (which takes up 12.4% of the data) and exclude the play records with response time less than 8 seconds and greater than 300 seconds (which takes up 5% of the data)

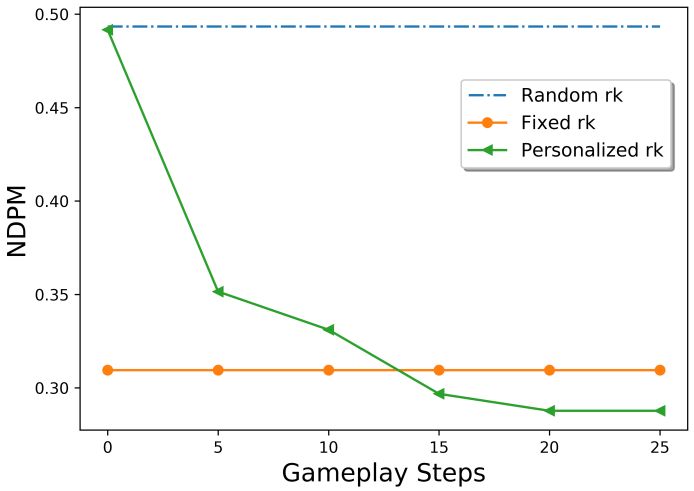


FIGURE 4.23: Difficulty ranking prediction error in terms of NDMP distance.

4.4 Summary

This chapter first proposed a general method for the determination of the cluster number and then introduced a clustering-based method to personalize the difficulty ranking. Experiments with real data from an online visual memory game have demonstrated the effectiveness of such personalization in improving the systems ability to predict the true difficulty ranking with a minimal number of gameplay observations. With this technique, we can now proceed to deal with the next research goal of dynamic difficulty adaptation (DDA) in order to present users with suitably challenging tasks. The next chapter will look at how this technique can be used to facilitate dynamic difficulty adaptation.

Chapter 5

Reinforcement Learning-based Difficulty Adjustment

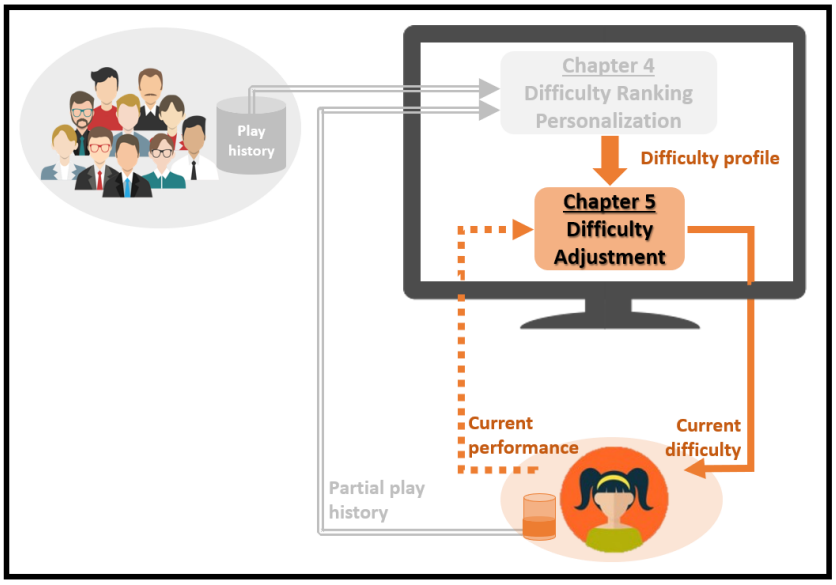


FIGURE 5.1: Problem structure of difficulty adjustment.

The previous chapter proposed a clustering-based method to identify the personalized difficulty ranking for each user. Given this personalized difficulty ranking, the next challenge is how to dynamically adjust the difficulty level for each user. This is the focus of this chapter (see Figure 5.1) [1]. A typical difficulty adaptation system [9, 15, 23] is shown in Figure 5.2, which follows an intuitive adaptation rule: present the user with an easier task if the current task is found to be too

[1]Section 5.1 is published as [143].

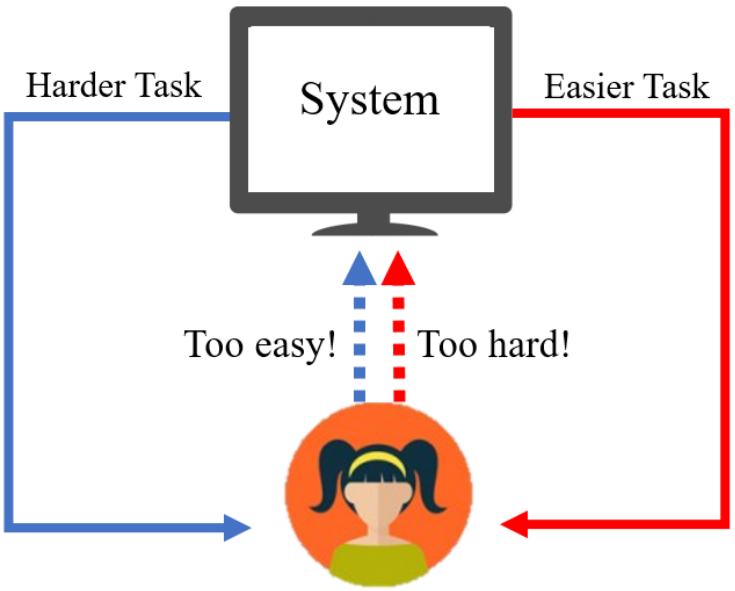


FIGURE 5.2: Difficulty adaptation system with human in the loop.

hard; give the user a harder task if the current one is too easy. To apply this rule, we first need to know which tasks are considered harder or easier for a particular user. This is the problem of personalized difficulty ranking and is exactly what has been solved in the previous chapter. However, after knowing which tasks are harder/easier, to actually perform difficulty adaptation, there is still an important question of how to make a selection among the several harder/easier questions. This is called the problem of difficulty adjustment. To solve this problem, some deterministic selection rules, like incremental update can be employed, which selects the question with the adjacent difficulty level (+/-1) at each adjustment step. However, the incremental update can result in very slow adaptation, especially when the question bank is large. For example, there are 100 question candidates in our experiment and only 25 interaction steps during the visual memory game play. With incremental update, it would take too many attempts to find the suitable difficulty levels for the user before the game is over. With the human in the loop, a responsive system must be able to make a correct selection within only a few steps or observations. To speed up, we can consider more intelligent rules like bisection, which repeatedly bisects the difficulty intervals and selects the one with the desired difficulty levels for further processing. In this way, the time complexity can be reduced from $O(N)$ to $O(\log(N))$ with N as the number of question candidates. However, this rule is still deterministic, thus making it sensitive to noise. If at one step, due to some stochastic variations in the user's performance or inaccurate

information in the difficulty ranking, the wrong sub-interval is selected, then this mistake would unavoidably lead to wrong difficulty levels. And real applications with varying visual memory profiles among different users contain lots of noise. In these cases, stochastic methods which are robust to noise, are probably more desirable choices compared to deterministic rules. For example, instead of directly selecting a certain kind of desired questions, we can maintain a probability of all the questions and increase the selecting probability of some desired questions. In this way, even with the interference of noise, there is still some chance of correct selection. However, despite the advantage of the stochastic adjustment, we are still faced with a critical question of how much increase should be made to the probability. Previous works using stochastic adaptation often rely on heuristic rules with some parameters to determine this [23], which can be hard to tune in practice. In addition, an important question of how to make stochastic adaptation converge to the optimal tasks has hardly been discussed in the previous works. This chapter seeks to address the challenges associated with stochastic difficulty adjustment.

The chapter is organized as follows. Section 5.1 proposes a new approach for stochastic difficulty adjustment based on reinforcement learning (RL), which can achieve fast and unbiased convergence with small batch size. In Section 5.2, the proposed difficulty adjustment approach is combined with the personalized difficulty ranking work in Chapter 4 to achieve dynamic difficulty adaptation in the Pals online visual memory game, an example of a human-in-the loop interactive system.

5.1 Algorithm: Bootstrapped Policy Gradient for Difficulty Adjustment

The selection of an appropriate question can be formulated as a sequential decision-making problem that can be solved using reinforcement learning algorithms. Related research works have applied RL for interaction optimization in Human Computer Interaction (HCI), where the RL agent interacts with the user to make some interactive decisions with the aim of enhancing user experience. For instance, Markov decision process (MDP) or partially observed Markov decision process (POMDP) frameworks have been used to support the sequential decision-making,

such as sequencing seven education concepts [11], choosing from two pedagogical strategies [144] and selecting from three display intervention messages [21]. In these applications, the action space is small, often less than 10. When it comes to larger action space, MDP/POMDP suffer from the curse of dimensionality. In intelligent tutoring systems, there could be hundreds or even thousands of candidates in the questions bank [12, 103, 115]. In these large action space cases, Multi-armed Bandit (MAB) or contextual MAB frameworks are usually employed as more scalable approaches [12, 145, 146]. The most similar work to that explored here is Maple (Multi-Armed Bandits based Personalization for Learning Environments) [23] which also formalizes difficulty adaptation in the framework of MAB. This work heuristically used the prior information of difficulty ranking to improve the efficacy for difficulty adjustment. However, despite the promising empirical results, no theoretical guarantee for convergence is provided and it is unclear whether the algorithm introduces bias to the optimal question. Generally, compared to the numerous studies of RL in other domains, like control systems or game-playing (Atari games, board games), the work of applying RL to benefit human-in-the-loop interactions is much more limited.

A major challenge in applying RL algorithms to real-world interactive application lies in sample efficiency. Value-based reinforcement learning methods [35, 147, 148], such as (deep) Q-learning, have been considered as sample efficient since it can train on off-policy data with experience replay technique [148]. Nevertheless, a suitable exploration policy [149–151], such as ϵ-greedy, Thompson sampling, is required in off-policy learning for efficient exploration. The trade-off between exploration and exploitation is a tricky issue [152]. As the action space becomes larger, efficient exploration becomes more challenging. Generally, the convergence of value function-based algorithms is not guaranteed [33, 153, 154]. Compared to the value-based method, policy gradient-based methods [32, 37, 155, 156] exhibit more stable convergence both in theory and in practice because these methods directly estimate the gradient of RL objective [33]. However, a major limitation for policy gradient methods is that they are severely sample inefficient due to the on-policy learning and high variance in gradient estimation [32, 33]. To achieve stable iterative policy optimization, a large batch size needs to be employed. For example, for continuous control task in Mujoco simulator [157], the common choice of batch size is often above 1000. However, responsive interactive applications cannot afford to use such large batch sizes as it will result in slow adaptation to the user. In fact,

instead of batch update, the incremental method, which updates the parameters immediately after receiving user feedback, is often used to ensure good user experience [12, 23, 146]. Therefore, this section investigates how to make the policy gradient method feasible and stable for small batch size update. Specifically, this work focuses on how policy gradient can be bootstrapped by prior information and then applied in the environments with short exploration horizon T and large action space A ($T \ll A$). This section makes a threefold contribution. Firstly, a general framework (BPG) was proposed for incorporating the prior information of action relations into policy gradient to achieve stable policy optimization even with small batch size. A sufficient condition was identified to guarantee unbiased convergence while employing BPG. Secondly, based on this framework, a BPG-based difficulty adaptation scheme was developed which can be applied in intelligent tutoring systems with large action space and short horizon. Thirdly, the generalization of BPG for multi-dimensional continuous action space in the general actor-critic reinforcement learning methods was studied.

5.1.1 Background

5.1.1.1 Problem Formulation

Consider a MAB framework defined by $\langle \mathcal{A}, \mathcal{R} \rangle$, where $\mathcal{A} = \{a_1, ..., a_A\}$ is a finite set of actions and $\mathcal{R} : \mathcal{A} \to \mathbb{R}$ is the reward function. The agent samples an action a_i from a stochastic policy $\pi_\theta(a) : \mathcal{A} \to [0, 1]$ parameterized by θ. The environment generates a reward r_{a_i} from a unknown probability $r_{a_i} \sim P(r|a_i)$, indicating how good the action is. The goal of the agent is to maximize the one-step MDP return as Equation 5.1.

$$\max_\theta J(\theta) = \max_\theta \mathbb{E}_{a_i \sim \pi_\theta}[r_{a_i}] \tag{5.1}$$

A commonly used policy is the softmax policy $\pi_\theta(a_i) = \frac{e^{w_i(\theta)}}{\Sigma_k e^{w_k(\theta)}}$ with $w \in \mathbb{R}^A$ as the softmax weights and thus $e^{w_i(\theta)} \propto \pi_\theta(a_i)$. (To simplify the notation, $w_i(\theta)$ will be frequently denoted as w_i). The softmax weights can be a more complex function $w(\theta, \phi)$, such as neural network, w.r.t θ and the action features $\phi(a_i)$.

The problem of difficulty adaptation can be formulated in the context of MAB by taking questions as actions and the suitability of a question for a user as the reward. Specifically, a setting with two actors is considered: a user who is interacting with

a system to complete a number of questions; and an agent selecting questions from a number of question candidates $\mathcal{A} = \{a_1, ..., a_A\}$ aiming to suitably challenge the user. At each time step, $t \in [1, ..., T](T \ll A)$, the agent selects a question a_i from question bank $\mathcal{A}$ and the user engages with the question and then the agent receives an observation of grade g_{a_i} and a scalar value of reward r_{a_i}. The grade g is negatively related to difficulty. If the grade is too high, the question is too easy for the user and vice verse. A target grade value $G \in \mathbb{R}$ is specified in advance to indicate the state of the user being suitably challenged. The reward indicates the suitability of this question for the user, which is measured by the distance of the current grade to the target grade $R_{a_i} = -|g_{a_i} - G|$. A ranking of questions $DR \in \mathbb{R}^M$ is known in advance. $DR(a_k) < DR(a_i)$ refers to task a_k is easier than a_i. In summary, the input is a target performance G and a known task difficulty ranking $DR, a \in \mathcal{A}$. The agent outputs a question to the user at each time step with the goal of keeping the user grade as close to the target value as possible, i.e. selecting the optimal action with the highest reward.

5.1.1.2 Policy Gradient

Based on the stochastic policy gradient theorem [156], the gradient of the objective function in Equation 5.1 can be reduced to a simple expectation, which can be obtained through sample-based estimates as shown in Equation 5.2. An intuitive understanding of this method is that the parameter is adjusted to update the exploration probability of an action based on the reward it receives. When an action leads to a high reward, the policy parameter will be adjusted to select this action more often.

$$\nabla_\theta J(\theta) = \mathbb{E}_{a_i \sim \pi_\theta}[r_{a_i} \nabla_\theta \log \pi_\theta(a_i)] \tag{5.2}$$

As mentioned earlier, the policy gradient method suffers from high variance in gradient estimation and thus necessities a large batch size for stable policy optimization. To better illustrate why large batch size is necessary, how the softmax weights are updated in Equation 5.2 is shown here. In the true gradient, the softmax weight of an action will be increased if and only it has a better-than-average reward, i.e. $\nabla_{w_k} J(\theta) = \pi_\theta(a_k)(r_{a_k} - \mathbb{E}_{a \sim \pi_\theta}[r_a])$. However, in the one-sample estimation of the gradient, as long as the sampled reward r_{a_i} is positive, the softmax weight of the sampled action a_i will always be increased, i.e.

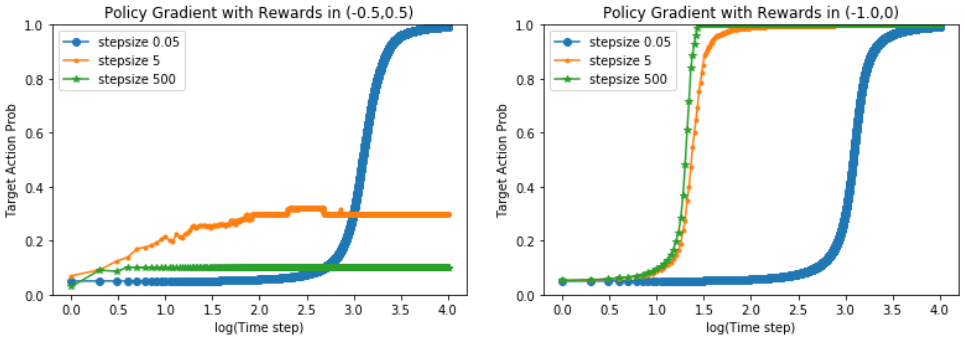
$\nabla_{w_i}\hat{J}(\theta)|_{a_i} = (1-\pi_\theta(a_i))r_{a_i}$, and all the other actions' weights will be always decreased, i.e. $\nabla_{w_j}\hat{J}(\theta)|_{a_i} = -\pi_\theta(a_j)r_{a_i}, a_j \neq a_i$. With infinite samples, all update inaccuracies will be canceled out and eventually, an accurate estimation of the gradient can be obtained, i.e. $\nabla_{w_k}J(\theta) = \mathbb{E}_{a_i\sim\pi_\theta}[\nabla_{w_k}\hat{J}(\theta)|_{a_i}]$. But in realistic scenarios, we do not have infinite samples to estimate one gradient step. Therefore, many researchers have studied how to reduce the variance in estimation so that a small number of samples can achieve an accurate estimation of the gradient [32, 33, 154, 155, 158, 159]. In these works, new score functions $f(a)$ are proposed to replace the raw reward r_a in the gradient estimation sample in Equation 5.2, i.e. $\nabla_\theta J(\theta) = \mathbb{E}_{a_i\sim\pi_\theta}[f(a_i)\nabla_\theta \log \pi_\theta(a_i)]$. The score functions $f(a_i)$ are often constructed by subtracting the reward with some baselines. If the baselines are independent of actions, i.e. $f(a_i) = r_{a_i} - B$, then despite the value change at the individual gradient estimation sample, the overall expectation of the gradient estimation samples remains the same because $\mathbb{E}_{a_i\sim\pi_\theta}[B\nabla_\theta \log \pi_\theta(a_i)] = 0$. In other words, adding any action-independent baseline will not introduce any bias to the gradient direction [36, 153]. In terms of the exact value of baseline, a common choice is the average reward $B = \mathbb{E}_{a\sim\pi_\theta}[r_a]$. In this way, the variance is reduced because at each gradient estimation sample, the sampled action's probability will be increased only when it receives a better-than-average reward instead of a positive reward as in the original case in Equation 5.2.

5.1.2 Sample Efficient Policy Gradient

5.1.2.1 Motivation

To apply policy gradient into the problem with short horizon T and large action space $\mathcal{A}$ ($T \ll A$), we examined the policy gradient method with the batch size equal to one and found that the method would fail in this scenario even with the above variance reduction scheme. Note that in the individual policy gradient estimation sample, the softmax weight of the sampled action is updated in one direction and all the other actions' updated in the opposite direction. And the above variance reduction scheme does not change this fact. As a result, the agent is still susceptible to being stuck at the sampled action. Specifically, if the sampled action has a better-than-average reward, i.e. $f(a_i) > 0$, only the sampled action's softmax weight will be increased and thus its probability is guaranteed to be enhanced. In

fact, if $f(a_i) > 0$, the probability of the sampled action over that of other actions is exponentially increasing at an rate of the step size α, since $\frac{\pi_\theta^{t+1}(a_i)}{\pi_\theta^{t+1}(a_k)} = \frac{\pi_\theta^t(a_i)}{\pi_\theta^t(a_k)} e^{\alpha x_i(a_k)}$, where $x_i(a_k) = \nabla_{w_i} \hat{J}(\theta)|_{a_i} - \nabla_{w_k} \hat{J}(\theta)|_{a_i} = f(a_i)(1 + \pi_\theta^t(a_k) - \pi_\theta^t(a_i))$ and $x_i(a_k) > 0$ if $f(a_i) > 0$. Hence, the step size needs to be kept very small when receiving positive score function values $f(a_i) > 0$. For the case with $f(a_i) < 0$, the issue of being stuck at the sampled actions can be alleviated since the agent does not increase a single action's softmax weight but increases for multiple actions, i.e. all the actions not sampled $a_k \neq a_i$. However, the policy gradient method would still be unfeasible in the problem with short horizon T and large action space A ($T \ll A$), due to the fact the number of exploration steps T needed in this method has to be greater than the action numbers A. As the method does not use any prior knowledge of the actions, it has to see all the actions, at least once, to decide which one is the best. These problems of the policy gradient method are demon-



(a) (b)

FIGURE 5.3: Policy gradient with batch size equal to one for (a) Problem 1 and (b) Problem 2.

strated using two toy problems. The two problem sets contain 20 actions with fixed rewards uniformly distributed from [-0.5,0.5] for problem 1 and [-1,0] for problem 2. Figures 5.3(a) and 5.3(b) show the probability of selecting the optimal action while applying incremental policy gradient to problems 1 and 2 respectively. The results show that for problem 1, the step size has to be kept small to avoid being stuck in sub-optimal actions. As a result, it needs nearly 10,000 steps to converge, even with this simple problem. The large step sizes can be used in problem 2 to achieve faster convergence but the convergence steps are bounded at the number of actions ($\log 20 \approx 1.3$), no matter how large the step size is.

To overcome these problems, a new method called Bootstrapped Policy Gradient (BPG) is introduced, which incorporates prior information of action relationship

into the policy gradient to bootstrap policy optimization. The proposed method can achieve stable and faster convergence to the target optimal action (without actually seeing all the actions) and can thus be applied in problems with the short horizon and large action space.

5.1.2.2 Bootstrapped Policy Gradient

Consider a piece of prior information which states certain actions are likely to have higher/lower reward than others. We will first discuss how to incorporate such prior information into policy gradient with unbiased convergence guarantee and then discuss how such information can be obtained in practice.

The key idea proposed here is to update the probability of *a set of actions* instead of a single action in the gradient sample. Let $\mathcal{X}_{a_i}^+$ denote *better action set*, which includes the actions that might be better than a_i and $\mathcal{X}_{a_i}^-$ denote a *worse action set*, which contains the worse actions than a_i. The bootstrapped policy gradient[2] formalized in Equation 5.3 increases the probability of the better action set $\widehat{\pi}_\theta^+(a_i) := \Sigma_{a_k \in \mathcal{X}_{a_i}^+} \pi_\theta(a_k)$ and decreases the probability $\widehat{\pi}_\theta^-(a_i) := \Sigma_{a_k \in \mathcal{X}_{a_i}^-} \pi_\theta(a_k)$ of the worse action set.

$$\tilde{\nabla}_\theta J(\theta) = \mathbb{E}_{a_i \sim \pi_\theta}[|r_{a_i}|(\nabla_\theta \log \widehat{\pi}_\theta^+(a_i) - \nabla_\theta \log \widehat{\pi}_\theta^-(a_i))] \tag{5.3}$$

Compared to traditional policy gradient, the proposed method enjoys several advantages. Firstly, in each gradient sample, the agent does not raise a single action's probability weights but that of a set of actions'. Thus it is more stable and less likely to be stuck with a certain action, regardless of the sign of the reward. This can also be shown in the softmax weights $\tilde{\nabla}_{w_k} \hat{J}(\theta) = \frac{\pi_\theta(a_k)}{\widehat{\pi}_\theta^+(a_i)} |r_{a_i}|, a_k \in \mathcal{X}_{a_i}^+$ and $\tilde{\nabla}_{w_k} \hat{J}(\theta) = -\frac{\pi_\theta(a_k)}{\widehat{\pi}_\theta^-(a_i)} |r_{a_i}|, a_k \in \mathcal{X}_{a_i}^-$, where the weight change direction no longer relies on whether the action is the sampled action and the sign of the reward. This property makes it possible for BPG to stably update policy even with a batch size of one. Secondly, in BPG the "worse" action's probability can be decreased without actually exploring it and the "better" action's probability can be increased before it has been selected. It is this property that makes it possible for BPG to find the

[2] In the case of $\widehat{\pi}_\theta(a_i) = 0$, the gradient update is set to be zero by letting $\widehat{\pi}_\theta(a_i)$ to be equal to a constant.

best action without exhaustively trying every action. In the interactive application, this means that the agent can eliminate some undesirable choices without actually exposing them to the users and use the limited exploration steps to focus on the more promising ones.

In spite of the promising properties of the BPG, an immediate question is how to ensure that the surrogate gradient can still lead to the target optimal action, given that the gradient direction has been altered. Notably, the performance of BPG is dependent on the "quality" of the "better/worse action set". Therefore, the next section will investigate what kind of constraints on "better/worse action set" are required for unbiased convergence.

5.1.2.3 Convergence Analysis

The original target action(s) is formally defined as $a_* := \arg\max r_a$. The goal is to make the surrogate gradient converge to a policy, where $\pi_\theta(a_k) = 0, \forall a_k \neq a_*$. For convenience, we define $\mathcal{A}_\theta^+ := \{\forall a_i | \widehat{\pi}_\theta^+(a_i) > 0\}$ and $\mathcal{A}_\theta^- := \{\forall a_i | \widehat{\pi}_\theta^-(a_i) > 0\}$. Then from Equation 5.3 we have:

$$\begin{aligned}\tilde{\nabla}_\theta J(\theta) &= \sum_{a_i \in \mathcal{A}_\theta^+} \pi_\theta(a_i)|r_{a_i}|\nabla_\theta \log \widehat{\pi}_\theta^+(a_i) \\ &\quad - \sum_{a_i \in \mathcal{A}_\theta^-} \pi_\theta(a_i)|r_{a_i}|\nabla_\theta \log \widehat{\pi}_\theta^-(a_i) \\ &= \sum_{a_i \in \mathcal{A}_\theta^+} \frac{\pi_\theta(a_i)|r_{a_i}|}{\widehat{\pi}_\theta^+(a_i)} \nabla_\theta \widehat{\pi}_\theta^+(a_i) - \sum_{a_i \in \mathcal{A}_\theta^-} \frac{\pi_\theta(a_i)|r_{a_i}|}{\widehat{\pi}_\theta^-(a_i)} \nabla_\theta \widehat{\pi}_\theta^-(a_i).\end{aligned} \tag{5.4}$$

The first equality uses the definition of expectation. The second equality uses the property of $\nabla_\theta \pi_\theta(a_i) = \pi_\theta(a_i) \nabla_\theta \log \pi_\theta(a_i)$.

We define $h_\theta^+(a_i) := \begin{cases} \dfrac{\pi_\theta(a_i)}{\widehat{\pi}_\theta^+(a_i)}|r_{a_i}|, & \widehat{\pi}_\theta^+(a_i) > 0 \\ 0, & \widehat{\pi}_\theta^+(a_i) = 0 \end{cases}$ (likewise for $h_\theta^-(a_i)$). Following these definitions, Equation 5.4 can be transformed as follows:

$$\begin{aligned}\tilde{\nabla}_\theta J(\theta) &= \sum_{a_i} h_\theta(a_i)\nabla_\theta \widehat{\pi}_\theta^+(a_i) - \sum_{a_i} h_\theta^-(a_i)\nabla_\theta \widehat{\pi}_\theta^+(a_i) \\ &= \sum_{a_i} h_\theta^+(a_i) \sum_{a_k \in \mathcal{X}_{a_i}^+} \nabla_\theta \pi_\theta(a_k) - \sum_{a_i} h_\theta^-(a_i) \sum_{a_k \in \mathcal{X}_{a_i}^-} \nabla_\theta \pi_\theta(a_k) \\ &= \sum_{a_k} \nabla_\theta \pi_\theta(a_k)\Big(\sum_{a_i \in \mathcal{X}_{a_k}^{+\prime}} h_\theta^+(a_i) - \sum_{a_i \in \mathcal{X}_{a_k}^{-\prime}} h_\theta^-(a_i)\Big),\end{aligned} \tag{5.5}$$

where $\mathcal{X}_{a_k}^{+\prime} := [\forall a_i | \mathcal{X}_{a_i}^+ \supseteq a_k]$ and $\mathcal{X}_{a_k}^{-\prime} := [\forall a_i | \mathcal{X}_{a_i}^- \supseteq a_k]$ are the *inverse set* of $\mathcal{X}_{a_k}^+$ and $\mathcal{X}_{a_k}^-$, which consists of all the actions whose *better(worse) action set* contains action a_k. The first equality uses the definition of $h_\theta^+(a_i)$ and $h_\theta^-(a_i)$. The second equality uses the definition of $\widehat{\pi}_\theta(a)$. The third equality reverses the order of i and k based on the definition of $\mathcal{X}(a)$ and $\mathcal{X}'(a)$. From above derivation, we can see the proposed policy improvement method in Equation 5.3 can be expressed in a form similar to the original policy gradient in Equation 5.2 by using a new score function estimator [3] $f_\theta(a)$ to replace r_a:

$$\begin{aligned}\tilde{\nabla}_\theta J(\theta) &= \sum_{a_k} f_\theta(a_k)\nabla_\theta \pi_\theta(a_k) \\ &= \mathbb{E}_{a_k \sim \pi_\theta}[f(a_k)\nabla_\theta \log \pi_\theta(a_k)]\end{aligned} \tag{5.6}$$

where $f_\theta(a_k) = \sum_{a_i \in \mathcal{X}_{a_k}^{+\prime}} h_\theta^+(a_i) - \sum_{a_i \in \mathcal{X}_{a_k}^{-\prime}} h_\theta^-(a_i)$.

We now examine if there is a certain special class of $f_\theta(a)$ which can make the surrogate gradient direction still converge to a_*. Note that if $f(a)$ is unrelated to θ, the condition for such unbiased convergence is straightforward, which is $\forall a \neq a_*, f(a_*) > f(a)$. However, when $f_\theta(a)$ is related to θ, it is not immediately obvious what the condition is, since it is hard to obtain the explicit expression of $\tilde{J}(\theta)$. Focusing on the case of softmax policy, a sufficient condition for unbiased convergence was identified in Theorem 5.1.1. The detail proof is shown in the Appendix C.

Theorem 5.1.1. (Surrogate Policy Gradient Theorem)
Given an action space $\mathcal{A} = \{a_1, .., a_A\}$, *a softmax exploration policy* $\pi_\theta(a_k) = \frac{e^{w_k(\theta)}}{\Sigma_i e^{w_i(\theta)}}$ *parameterized by* θ, *and a target action set* a_*. *Consider a surrogate policy gradient for policy optimization defined by a score function* $f_\theta(a)$: $\tilde{\nabla}_\theta J(\theta) =$

[3] $f_\theta(a)$ is a function on $\mathcal{X}^+$ and $\mathcal{X}^-$ and should be denoted as $f_{\theta,\mathcal{X}^+,\mathcal{X}^-}(a)$. These variables are dropped to simplify notation.

$\Sigma_{a_k} f_\theta(a_k) \nabla_\theta \pi_\theta(a_k)$, *then for the policy optimization to converge at the target action set* a_*, *i.e.* $\pi_\theta(a) = 0, \forall a \neq a_*$, *a sufficient condition C.1 is:* $\forall a \neq a_*$

- $f_\theta(a_*) \geq f_\theta(a), \forall \theta$
- $f_\theta(a_*) > f_\theta(a), \forall \theta \in \{\theta | 0 < \pi_\theta(a) < 1 \& \pi_\theta(a_*) \neq 0\}$

In other words, Theorem 5.1.1 gives a class of score function $f_\theta(a)$ which can guarantee the surrogate gradient to the target action a_*. This class of score function needs to meet two conditions:

1) the values of $f_\theta(a)$ at the target optimal actions a_* are always better or equal to that of all the other actions, regardless of θ; and
2) the equality only exists at certain space of θ, where $\pi_\theta(a) = 0$ or $\pi_\theta(a) = 1$ or $\pi_\theta(a_*) = 0$.

In fact, the previous method with action-independent baseline [32, 155] is a special case of this theorem, since its score function $f(a_i) = r_{a_i} - B$ satisfies the above conditions. Unlike previous works that endeavor to maintain unbiased gradient estimation [32, 33, 154, 155, 158, 159], this work proved it is legitimate to use biased gradient, as long as the proposed sufficient condition C.1 is met.

5.1.3 Difficulty Adjustment

The previous section points out a sufficient condition for BPG to achieve unbiased convergence. This section discusses how to obtain better/worse action sets $\mathcal{X}_a^+$ and $\mathcal{X}_a^-$ to actually satisfy this sufficient condition. Note in practice, we do not know exactly which actions are better than which, since if we have this information, the problem would already be solved. Thus one can only work with inaccurate "better/worse action sets". Therefore, an interesting question is whether an inaccurate "better/worse action sets" can lead to sufficient condition C.1 to be met and if so how.

In the case of difficulty adaptation, there happens to be a convenient way to construct *approximate better/worse action sets* from prior information of difficulty

ranking. Specifically, if a question is observed to be too easy or too hard for the user, then those questions which are even easier or harder than the current one can be considered as "worse" actions; and in contrast those questions which are harder or easier than the current one can be considered as "better" actions. Following the problem formulation of difficulty adaptation described in Section 5.1.1.1, the approximate "better action set" and "worse action set" are expressed as follows:

$$\mathcal{X}_a^+ := \begin{cases} \forall a_k | DR(a_k) > DR(a), & g_a > G \\ \forall a_k | DR(a_k) < DR(a), & g_a < G \\ \emptyset, & g_a = G; \end{cases} \tag{5.7}$$

$$\mathcal{X}_a^- := \begin{cases} \forall a_k | DR(a_k) < DR(a), & g_a > G \\ \forall a_k | DR(a_k) > DR(a), & g_a < G \\ \forall a_k | DR(a_k) \neq DR(a), & g_a = G. \end{cases} \tag{5.8}$$

Although the information contained in above better/worse action sets is not accurate, the BPG with these sets can still guarantee unbiased convergence, because the corresponding $f_\theta(a)$ indeed satisfies the condition C.1. The proof is as follows.

First, some notations which will be used in the proof are presented. We define $\mathcal{A}_L := \{a | g_a > G\}$ and $\mathcal{A}_R := \{a | g_a < G\}$ which denote the questions which are too easy or too hard for the user respectively. And the questions which are suitable challenging for the user is denoted as $\mathcal{A}_M := \{a | g_a = G\}$. Then $\mathcal{A}_M$ is the target optimal action set.

Based on the definition of inverse set, the corresponding inverse sets of the proposed better/worse action sets are in Equation 5.9 and (5.10) respectively.

$$\mathcal{X}_a^{+'} = \begin{cases} \mathcal{A}_R \cup \{\forall a_k | DR(a_k) < DR(a)\}, & a \in \mathcal{A}_L \\ \mathcal{A}_L \cup \{\forall a_k | DR(a_k) > DR(a)\}, & a \in \mathcal{A}_R \\ \mathcal{A}_L \cup \mathcal{A}_R & a \in \mathcal{A}_M; \end{cases} \tag{5.9}$$

$$\mathcal{X}_a^{-'} = \begin{cases} \mathcal{A}_M \cup \{\forall a_k | a_k \in \mathcal{A}_L \& DR(a_k) > DR(a)\}, & a \in \mathcal{A}_L \\ \mathcal{A}_M \cup \{\forall a_k | a_k \in \mathcal{A}_R \& DR(a_k) < DR(a)\}, & a \in \mathcal{A}_R \\ \emptyset, & a \in \mathcal{A}_M. \end{cases} \tag{5.10}$$

Note that the optimal actions have larger inverse better action sets $\mathcal{X}_a^{+'}$ and smaller inverse worse action sets $\mathcal{X}_a^{-'}$ than other actions, i.e. $\mathcal{X}_{a_*}^{+'} \supseteq \mathcal{X}_{a_k}^{+'}$ and $\mathcal{X}_{a_k}^{-'} \supseteq \mathcal{X}_{a_*}^{-'}$. Therefore, $\forall a_k \in \mathcal{A}_R$,

$$\begin{aligned}
& f_\theta(a_*) - f_\theta(a_k) \\
=& \sum_{a_i \in \mathcal{X}_{a_*}^{+'} \setminus \mathcal{X}_{a_k}^{+'}} h_\theta^+(a_i) + \sum_{a_i \in \mathcal{X}_{a_k}^{-'} \setminus \mathcal{X}_{a_*}^{-'}} h_\theta^-(a_i) \\
=& h_\theta^+(a_k) + h_\theta^-(a_*) + \sum_{a_i \in \mathcal{X}_{a_*}^{+'} \setminus \mathcal{X}_{a_k}^{+'} \cap \mathcal{X}_{a_k}^{-'} \setminus \mathcal{X}_{a_*}^{-'}} (h_\theta^+(a_i) + h_\theta^-(a_i)) \\
\leq& h_\theta^+(a_k) + h_\theta^-(a_*).
\end{aligned}$$

The first equality uses the definition of $f_\theta(a)$ and complementary set: $\mathcal{X}_{a_*}^{+'} \setminus \mathcal{X}_{a_k}^{+'} = \{\forall a_i | a_i \in \mathcal{X}_{a_*}^{+'} \& a_i \notin \mathcal{X}_{a_k}^{+'}\}$ and $\mathcal{X}_{a_k}^{-'} \setminus \mathcal{X}_{a_*}^{-'} = \{\forall a_i | a_i \in \mathcal{X}_{a_k}^{-'} \& a_i \notin \mathcal{X}_{a_*}^{-'}\}$. Following Equations 5.9 and 5.10, $\forall a_k \in \mathcal{A}_R$, $\mathcal{X}_{a_*}^{+'} \setminus \mathcal{X}_{a_k}^{+'} = \{\forall a_i | a_i \in \mathcal{A}_R \& DR(a_i) <= DR(a_k)\}$ and $\mathcal{X}_{a_k}^{-'} \setminus \mathcal{X}_{a_*}^{-'} = \mathcal{A}_M \cup \{\forall a_i | a_i \in \mathcal{A}_R \& DR(a_i) < DR(a_k)\}$. Thus, $\mathcal{X}_{a_*}^{+'} \setminus \mathcal{X}_{a_k}^{+'} \cap \mathcal{X}_{a_k}^{-'} \setminus \mathcal{X}_{a_*}^{-'} = \{\forall a_i | a_i \in \mathcal{A}_R \& DR(a_i) < DR(a_k)\}$, which leads to the second equality.

Based on this derivation and the fact that $h_\theta^+(a) \geq 0$ and $h_\theta^-(a) \geq 0$, we immediately have $f_\theta(a_k) \leq f_\theta(a_*), \forall a_k \in \mathcal{A}_R$. The case for $\forall a_k \in \mathcal{A}_L$ can be proven in a similar way. Therefore, it has been shown that $f_\theta(a)$ satisfies the first condition of having maximum value at the target optimal actions. Moreover, if $\pi_\theta(a_*) \neq 0$ and $\pi_\theta(a_k) \neq 0$, then $\widehat{\pi}_\theta^+(a_k) > 0$ and $\widehat{\pi}_\theta^-(a_*) > 0$. Combined with the definition of $h_\theta^+(a)$ and $h_\theta^-(a)$, we have $h_\theta^+(a_k) + h_\theta^-(a_*) = \frac{\pi_\theta(a_k)}{\widehat{\pi}_\theta^+(a_k)} |r_{a_k}| + \frac{\pi_\theta(a_*)}{\widehat{\pi}_\theta^-(a_*)} |r_{a_*}| > 0$, if $r_{a_k} \neq r_{a_*}$. Thus we have, $\forall a_k \in \mathcal{A}_R$, $f_\theta(a_k) < f_\theta(a_*)$. The case for $\forall a_k \in \mathcal{A}_L$ can be verified in a similar way. Therefore, we arrive at the conclusion that $f_\theta(a)$ also satisfies the second condition.

In summary, it has been shown that with the proposed better/worse action sets in Equations 5.7 and 5.8, condition C.1 is met and thus the proposed BPG-based difficulty adjustment approach is guaranteed to converge at the target optimal action. The overall difficulty adaptation algorithm is shown in Table 3. In addition, although the algorithm discussed here simultaneously increases the probability of the better action set and decreases that of the worse action set, other methods focusing on one direction adjustment can also be analyzed in BPG framework by simply setting $\mathcal{X}_a^+$ or $\mathcal{X}_a^-$ to be $\emptyset$. Such an example (Maple-like BPG) can be found in Section 5.1.5.

Algorithm 3: Algorithm of BPG-based online difficulty adjustment

Input: target $G \in \mathbb{R}$, difficulty ranking $D_a \in \mathbb{R}^A$
Output: next question a_i for each user at each time step
Initialize: policy parameters $\theta_k = 0, k = 1, ..., A$
For each time step:
  Sample a question $a_i \sim \pi_\theta$ from current policy
  Get grade g_a from user
  Obtain *related action sets* $\mathcal{X}_{a_i}^+$ and $\mathcal{X}_{a_i}^-$ with Eq. 5.7 and Eq. 5.8
  Update parameters $\theta = \theta + \alpha \nabla_\theta J$ with Eq. 5.3
  Compute new policy $\pi_\theta = \text{softmax}(\theta)$

5.1.4 Generalization in Actor-Critic Methods

Previous section focused on how to employ BPG in difficulty adjustment problem. This section discusses how BPG can be applied to the general reinforcement learning problems beyond difficulty adaptation. One challenge in the generalization of BPG is the issue of obtaining the better/worse action set without prior information. It turns out such information is surprisingly easy to obtain in actor-critic reinforcement learning methods. Actor-critic methods are a family of RL algorithms which combine the strength of both value-based methods and policy gradient methods. In these algorithms, a value function of $Q(a)$ (critic) is learned to indicate the goodness of each action and provides guidance for the policy (actor) optimization. Therefore, the information about whether an action might be better/worse than another is exactly what we can expect the critic to contain. Moreover, although the previous formulation is derived for discrete action space, the idea of increasing better action set and decreasing worse action set can also be used for multi-dimensional continuous action space. The continuous action case turns out to be very similar to the difficulty adaptation problem. The absolution value of the action contains a natural ranking and $\nabla_a Q(a)$ denotes whether the action value is too high or too low. Therefore, the better/worse action set in continuous action domain can be defined in a similar way with Equations 5.7 and 5.8:

$$\mathcal{X}_a^+ = \begin{cases} (a, \infty), \nabla_a Q(a) > 0 \\ (-\infty, a), \nabla_a Q(a) < 0 \end{cases} \quad \mathcal{X}_a^- = \begin{cases} (-\infty, a), \nabla_a Q(a) > 0 \\ (a, \infty), \nabla_a Q(a) < 0. \end{cases}$$

Following above definitions on $\mathcal{X}_a^+$, $\mathcal{X}_a^-$ and taking $|r_a|$ as $|\nabla_a Q(a)|$ in Equation

5.3, the continuous BPG is proposed as in Equation 5.11:

$$\tilde{\nabla}_\theta J(\theta) = \mathbb{E}_{a\sim\pi_\theta}[\nabla_\theta \log \tilde{\pi}_\theta(a) \nabla_a Q(a)] \tag{5.11}$$

where $\tilde{\pi}_\theta(a) := \frac{1-F(a)}{F(a)}$, $F(a) := [\mathbb{P}(x^i < a^i), x \sim \pi_\theta, i = 1, ..., D]$ is a vector and each component stands for cumulative distribution function (cdf) at each action dimension a^i. Note that with $a \in \mathbb{R}^D$ and $\theta \in \mathbb{R}^N$, $\nabla_\theta \log \tilde{\pi}_\theta(a_i) \in \mathbb{R}^{N\times D}$. The proposed continuous BPG is similar to deterministic policy gradient (DPG) [160]: $\nabla_\theta J(\theta) = \nabla_\theta \mu_\theta \nabla_a Q(a)|_{a=\mu_\theta}$, where $\mu_\theta \in \mathbb{R}^D$ is a deterministic policy parameterized by θ, as both methods can make use of the information of $\nabla_a Q(a)$ to improve sample efficiency. Specifically, given a multivariate Gaussian policy $\mathcal{N}(\mu, \sigma)$ and $\theta = [\mu, \sigma]$, we have $\nabla_{\mu^i} \log \tilde{\pi}_\theta(a^i) = \frac{\pi_\theta(a^i)}{F(a^i)} + \frac{\pi_\theta(a^i)}{1-F(a^i)} > 0$ for $i = 1, ..., D$. Hence, similar to DPG where $\nabla_{\mu^i}(\mu^i_\theta) = 1$ for $\theta = [\mu]$, continuous BPG also moves the policy in the direction of the gradient of Q and converges at the places of $\nabla_a Q(a) = 0$. However, one common limitation for continuous BPG and DPG is the local optimal issue in the case of a non-convex Q function (e.g. neural network), due to the dependence on $\nabla_a Q(a)$. DPG is a deterministic policy and only focuses on the area of $a = \mu_\theta$. Continuous BPG, on the other hand, is stochastic and can incorporate the $\nabla_a Q(a)$ information even when $a \neq \mu_\theta$. Thus, continuous BPG might have the potential to alleviate this local optimal problem, given that it can make use a wider range of $\nabla_a Q(a)$. A more rigorous convergence analysis and detailed comparison between continuous BPG and DPG will be conducted in the future work.

In practice, the critic function $Q(a)$ is usually obtained using a function approximator $Q^w(a)$. Generally, this replacement could affect the gradient direction. However, similar to traditional stochastic and deterministic policy gradient [156, 160], a family of compatible function approximator $Q^w(a)$ for bootstrapped policy gradient is identified in Theorem 5.1.2, such that substituting $Q^w(a)$ into Equation 5.11 will not affect the gradient.

Theorem 5.1.2. (Compatible function approximation) *A function approximator $Q^w(a)$ is compatible with a bootstrapped policy $\tilde{\nabla}_\theta J(\theta) = \mathbb{E}_{a_i\sim\pi_\theta}[\nabla_\theta \log \tilde{\pi}_\theta(a_i) \nabla_a Q^w(a)|_{a_i}]$, if*

(1) $\nabla_a Q^w(a)|_{a_i} = \nabla_\theta \log \tilde{\pi}_\theta(a_i)^T w$; *and*

(2) w *minimizes the mean-squared error, i.e.* $\min_w MSE(\theta, w) = \mathbb{E}[\epsilon(\theta, w)^T \epsilon(\theta, w)]$, *where* $\epsilon(\theta, w) = \nabla_a Q^w(a)|_{a_i} - \nabla_a Q(a)|_{a_i}$.

Proof. If w minimizes the MSE then the gradient of ϵ^2 w.r.t w must be zero. We then use the fact that, by condition (1), $\nabla_w \epsilon(\theta, w) = \nabla_\theta \log \tilde{\pi}_\theta(a_i)$,

$$\begin{aligned} 0 &= \nabla_w MSE(\theta, w) \\ &= \mathbb{E}[\nabla_\theta \log \tilde{\pi}_\theta(a_i)\epsilon(\theta, w)] \\ &= \mathbb{E}[\nabla_\theta \log \tilde{\pi}_\theta(a_i)(\nabla_a Q^w(a)|_{a_i} - \nabla_a Q(a)|_{a_i})]. \end{aligned} \tag{5.12}$$

Thus, $\mathbb{E}[\nabla_\theta \log \tilde{\pi}_\theta(a_i)\nabla_a Q^w(a)|_{a_i}] = \mathbb{E}[\nabla_\theta \log \tilde{\pi}_\theta(a_i)\nabla_a Q(a)|_{a_i}]$. □

For any stochastic policy $\pi_\theta(a)$, there always exists a compatible function approximator of the form $Q^w(a) = \phi(a)^T w$ with action features $\phi(a) := \nabla_\theta \log \tilde{\pi}_\theta(a) a^T$ and parameters w. Although a linear approximator is not effective for predicting action values globally, it serves as a useful local critic to guide the parameter update direction [160]. Regarding the condition (2), in theory, we need to minimize the mean square error between the gradient of $Q(a)$ and $Q^w(a)$. Since the true gradient $\nabla_a Q(a)$ is difficult to obtain, in practice, the parameter w is learned using the standard policy evaluation method, like Sarsa or Q-learning. The detail discussion regarding this issue can be found in [160]. In the experiment section 5.1.5, a concrete example of how to apply the bootstrapped policy gradient for the continuous-armed bandit is provided.

5.1.5 Experimental Results

5.1.5.1 Difficulty Adaptation

BPG was compared with five other difficulty adjustment methods:

- *Random method*: it always randomly selects questions ;
- *Bisection method*: a deterministic approach which repeatedly bisects the difficulty interval and then selects a subinterval which may contain the ideal difficulty level for further processing;
- *Policy gradient (PG)*: it updates policy based on Equation 5.2;
- *Maple method*: it heuristically increases the softmax weights of harder questions when a task is too easy for this user (i.e. $w = \alpha_1 e^{g_a - G} w, g_a > G$),

and decreases the softmax weights of harder questions otherwise (i.e. $w = \alpha_2 e^{g_a - G} w, g_a \leq G$) [23] with α_1 and α_2 as parameters; and

- *Maple-like BPG* (BPG_mpl)[4]: it uses the bootstrapped policy gradient in Equation 5.3, but the better/worse action sets defined following the rules used in Maple. Specifically, when the question is too easy, the harder questions are considered as the better action, i.e. $\forall a_i \in \mathcal{A}_L$, $\mathcal{X}_{a_i}^+ := \{\forall a_k | DR(a_k) > DR(a_i)\}$, $\mathcal{X}_{a_i}^- = \emptyset$; otherwise, harder questions are regarded as worse actions, i.e. $\forall a_i \notin \mathcal{A}_L$, $\mathcal{X}_{a_i}^+ := \emptyset$, $\mathcal{X}_{a_i}^- := \{\forall a_k | DR(a_k) > DR(a_i)\}$.

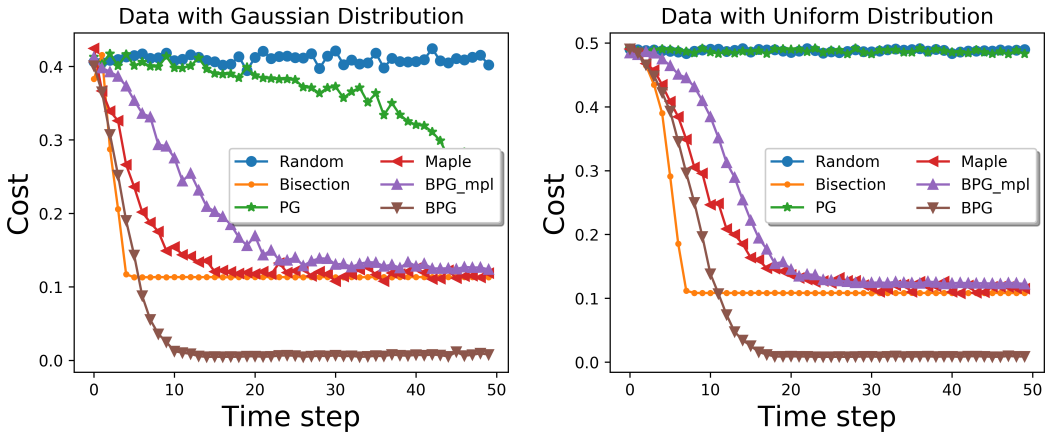
A parameter sweep on step size is performed for all the approaches.

The data is generated following a similar manner in [23, 146]. The user performance is measured by grade g, which is positively related to the probability of a user answering a task correctly. Given a target grade value G, which indicates the best user experience, the goal of the difficulty adjustment algorithm is to select the questions to keep the user's performance at the target grade as close as possible. Each user's ability is modeled by a competence level sl. Each question is modeled by a difficulty level ql. Given a pair of difficulty level and competence level, the grade the user may receive after answering this question is computed based on Item Response Theory [161]: $g = \beta \frac{1}{1+e^{\gamma(ql-sl)}} + (1-\beta)\varepsilon$. The parameters are set to be $\gamma = 1$ and $\beta = 0.1$. Note that β controls the amount of random noise $\varepsilon \sim \mathcal{U}[0, 1)$ in the reward. In this setting, when the question difficulty ql matches exactly with user ability sl, the grade is 0.5 and thus the target grade G is set to be 0.5 in the experiment. Two kinds of distributions are considered while generating the user competency levels sl and question difficulty levels ql: the uniform distribution $\mathcal{U}\{1, 200\}$ and Gaussian distribution $\mathcal{N}(100, 20)$. This experiment considered 500 users interact with the agent and each completes 50 questions. There are 1,000 possible question for selection (i.e. $T = 50, A = 1,000$).

Figure 5.4 shows the average cost at each time step for the different adjustment approaches. The cost is computed as $-r = |g - G|$, which indicates the distance to the optimal user experience. As expected, policy gradient method with a batch

[4]Note that the softmax weights update direction in Maple-like BPG, in terms of increasing or decreasing, is same with Maple. But unlike Maple in which the specific update amount is decided in a heuristic manner with several tunable parameters, Maple-like BPG uses BPG framework in Equation 5.3 to determine the update amount. The reason for employing Maple-like BPG as a baseline is to demonstrate how to use BPG framework to analyze other adjustment scheme besides the one proposed here.

size of one fails in this problem with $T \ll A$. In fact, its performance is almost as worse as random exploration. The proposed method as well as Bisection, Maple-like methods (Maple, Maple-like BPG) can quickly converge within 10 to 20 exploration steps. However, the proposed method leads to significantly lower cost than all the other methods. Bisection method uses deterministic policy, which makes it sensitive to the noise in the reward. As for the computation (see Table 5.1), the rule-based method achieve fastest performance with around 0.001 milliseconds for one adaptation step. The stochastic method achieve similar running time of about 0.01 milliseconds, which is still fast enough for supporting real-time responsive system.



(a) (b)

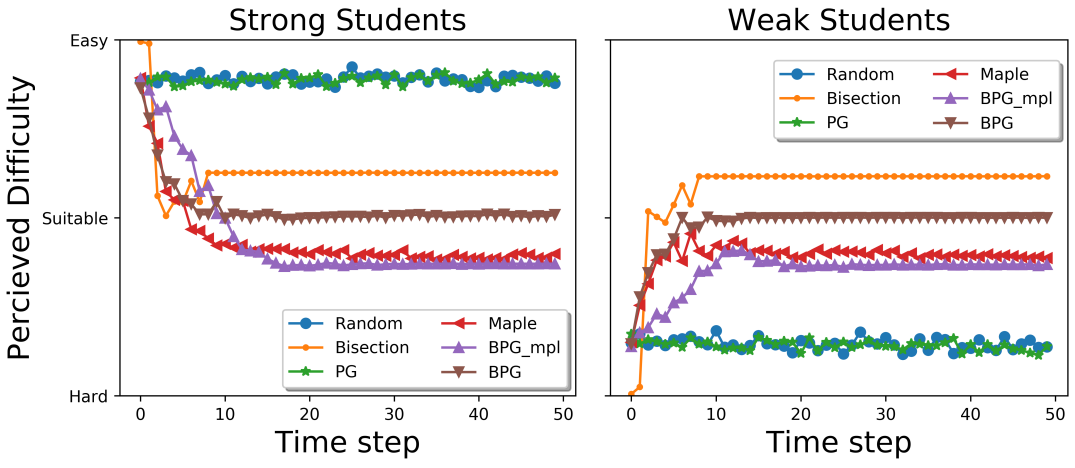
FIGURE 5.4: Comparison of adaptation methods for difficulty adjustment for data with a) Gaussian and (b) Uniform distributions.

TABLE 5.1: Average running time of an adaptation step (in milliseconds).

Random	Bisection	PG	Maple	BPG_mpl	BPG
0.010	0.001	0.013	0.012	0.013	0.013

Regarding the maple-like methods, it is not immediately obvious why they seem to fail to converge at the optimal actions as they follow intuitively reasonable rules to update the stochastic policy. This phenomenon is investigate in the following experiment. In particular, the agents' difficulty adjustment behavior for strong students (with top 25% competency levels) and weak students (with last 25% competency levels) were investigated. The users' perceived difficulty indicated by the users' grade is shown in Figure 5.5. The results show that the questions

generated by random selection and policy gradient are always too easy for the strong students (i.e. grades are close to 1) and too hard for the weak students (i.e. grades are close to zero). After about 10 adjustment steps, the proposed method can select questions to challenge students at a suitable level (grades are at the exact target value of 0.5). However, the Maple-like methods seem to favor harder questions by keeping the users' grades below 0.5. To explain this behavior, the score function $f_\theta(a)$ of Maple-like BPG was examined and the condition C.1 was found to be violated in Maple-like BPG. In fact, it meets the first part but violates the second part of the condition. Specifically, in its score function, the optimal action a_{R_*} in set $\mathcal{A}_R$ always has the same value with a_*, even if $0 < \pi_\theta(a_{R_*}) < 1$ and $\pi_\theta(a_*) \neq 0$. Because $f_\theta(a_*) - f_\theta(a_{R_*}) = h_\theta^-(a_*) = \dfrac{\pi_\theta(a_*)}{\widehat{\pi}_\theta^-(a_*)}|r_{a_*}|$. Given that the reward at target optimal question $|r_{a_*}|$ is close to zero in this problem, $f_\theta(a_{R_*})$ and $f_\theta(a_*)$ have similar values. In other words, the agent cannot differentiate a_* from a_{R_*}. It appears not all the heuristic-based difficulty adjustment schemes can converge towards optimal actions. To ensure unbiased selection, it is crucial to check whether the corresponding score function meets the sufficient condition C.1. In fact, it has been found that while using heuristic rules to construct better/worse action set, only the first condition is generally true, and one should pay particular attention to check if the second condition is met.



(a) (b)

FIGURE 5.5: Perceived difficulty for (a) stronger and (b) weaker students.

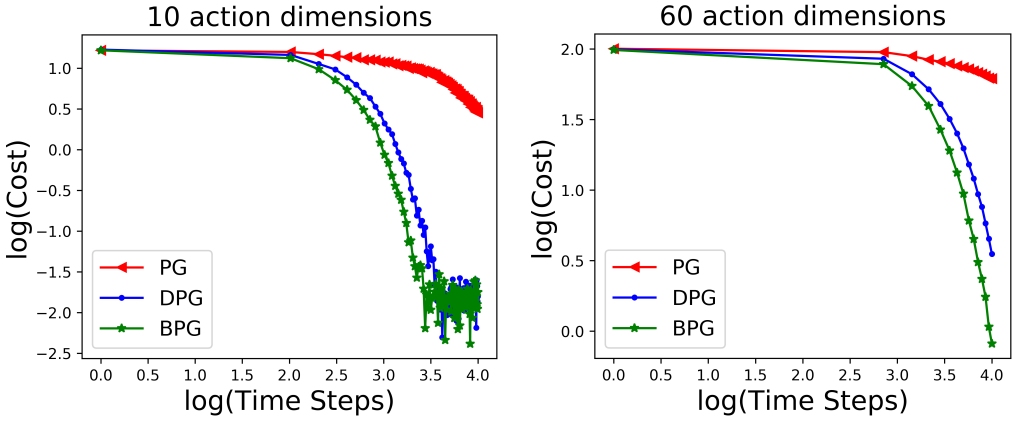
5.1.5.2 Continuous-armed Bandit

This experiment applied BPG in multidimensional continuous action domain. Specifically, the same continuous-armed bandit problem proposed in [160] was considered. A high-dimension quadratic cost function is considered in this problem, and is defined as $-r(a) = \beta(a - a_*)^T C(a - a_*) + (1 - \beta)\epsilon$, where $a_* = [4, ..., 4]^T$, $\beta = 0.99$ controls the amount of random noise $\varepsilon \sim \mathcal{U}[0, 1)$ in the reward and C is a positive definite matrix with eigenvalues of 0.1. Two systems were considered with action dimensions of 10 and 60 respectively, i.e. $a \in \mathbb{R}^{10}$, $a \in \mathbb{R}^{60}$. The performance of continuous BPG was compared with other well-established policy optimization methods such as deterministic policy gradient (DPG) [160] and stochastic policy gradient (REINFORCE) [155]. Same as in [160], the critic functions $Q^w(a)$ were estimated by linear regression from the features to the costs. The features [5] used are $(a - \mu)$. With the batch size twice that of the action dimension, the actor and the critic were updated per batch. A parameter sweep over all the parameters of step-size and variance was performed. Figure 5.6 shows the performance with the best parameter for each algorithm. From these results, we can see that with the 10 action dimensions, BPG outperforms stochastic policy gradient (PG) and achieves similar performance with DPG. As the problem becomes more challenging with higher dimension, the performance in BPG is better than all the other methods including DPG. Note that the control of complex system often involves high dimension of freedom degree. The Humanoid robot in Mujoco simulator [157] has 17 action dimensions. And the real world systems have even higher controlled degree of freedom, such as human body, which have over 600 muscles and 200 bones [162]. This result shows a promising application to explore high dimensional space.

5.1.6 Discussion

This work applies reinforcement learning to address the issue of difficulty adaptation with the goal of presenting users with suitably challenging tasks. To overcome the problem of sample inefficiency in policy gradient methods, a framework of the bootstrapped policy gradient (BPG) algorithm was presented, which can exploit the prior knowledge of action relationship to achieve stable policy optimization

[5] We also run experiment with corresponding compatible feature for BPG but the results do not improve.



(a) (b)

FIGURE 5.6: Comparison of policy gradient methods for the continuous-armed bandit with (a) 10 action dims, and (b) 60 action dims.

even with small batch size. The key idea is to increase the probability of better action set and decrease the probability of worse action set at gradient estimation sample. The BPG-based difficulty adaptation approach is able to achieve fast convergence in a challenging environment with short horizon and large discrete action space ($T \ll A$). On the theoretical front, unlike other heuristic-based difficulty methods, a rigorous theoretical justification was provided to guarantee that the proposed difficulty adjustment scheme can converge to the target optimal action. In fact, the sufficient unbiased convergence condition identified in the theoretical analysis can shed some light on why some seemly reasonable heuristic-based difficulty adjustment schemes sometimes fail. This is because the corresponding score function of these rules satisfies the first requirement of the sufficient condition but violates the second.

Several points should be noted when applying the proposed difficulty adaptation method to real-world interactive applications. The experiment given here used the relationship between the user's grade and the target grade to infer whether the current question is too easy or too hard for the user. For applications where target grade is unavailable, other indicators like the user's error rate or reaction time can be used to infer this information. In fact, the next section shows the application of BPG in the visual memory game, for which the memorization time is used as the measure of perceived difficulty (see Section 5.2).

The generalization of BPG to general reinforcement learning problems with no prior information available has also been discussed. In particular, a link between BPG and actor-critic methods was revealed by using the critic function to provide prior information for BPG. A continuous BPG for multi-dimensional continuous action domain was presented and the effectiveness of which has been demonstrated through the continuous-armed bandit problem. So far, the generalization of BPG was discussed in the context of MAB. Applying the idea of increasing the probability of the better action set and decreasing the probability of the worse action set in MDP is a future research direction.

5.2 Application in Visual Memory Game for Dynamic Difficulty Adaptation

5.2.1 Machine Learning-based Difficulty Adaptation

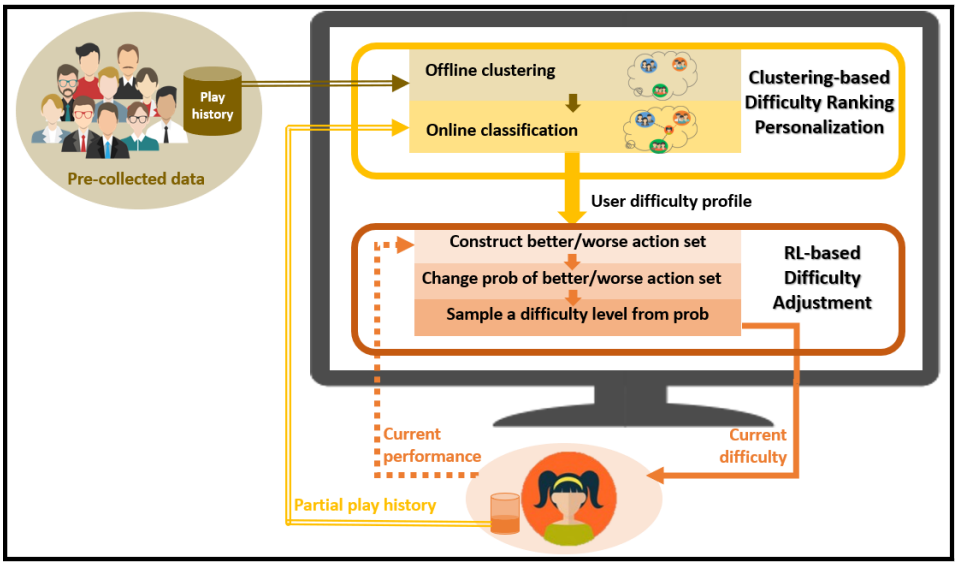


FIGURE 5.7: Overview structure of ML-based dynamic difficulty adaptation combining difficulty ranking personalization and stochastic difficulty adjustment.

The previous section proposed a method (BPG) to automatically adjust task difficulty to match user ability. BPG assumes the difficulty ranking is known in advance. Regarding how to obtain such prior information, Chapter 4 proposed a solution of clustering-based difficulty ranking personalized approach. Therefore,

Combining these two methods leads to a complete machine learning-based algorithm for dynamic difficulty adaptation (see Figure 5.7). Specifically, first, the Curvature-based clustering is employed on pre-collected performance data to identify the different difficulty profiles. Next, during the online interaction with users, the most appropriate difficulty profile is updated every a few steps (e.g. 5 steps) based on past play history and selected as the personalized difficulty ranking. At each time step, the the better/worse action sets are constructed based on the updated personalized difficulty ranking and the user's current performance feedback and then the policy parameters are adjusted accordingly to increase the probability of the better action set and decrease the probability of the worse action set following BPG. Finally, the next task is sampled from the probability with updated parameters. The detail of the full DDA algorithm is shown in Algorithm 4.

Algorithm 4: Algorithm for dynamic difficulty adaptation combining difficulty ranking personalization and stochastic difficulty adjustment

Offline stage
Preprocess the performance data: $DR = preprocessing(X)$
Determine the cluster number: $k = curvatureMethod(DR)$
Obtain the difficulty ranking candidates by clustering
Online stage
For each time step t:
  Sample a question $a_i \sim \pi_\theta$ from current policy
  Get grade g_a from user
  Obtain *better/worse action sets* based on difficulty ranking
  Update policy with BPG
  For every 5 time steps ($t\%5 == 0$):
    Update the personalized difficulty ranking with latest play history

In the next section, the Pals online visual memory game platform is taken as an example to demonstrate how ML-based DDA method can be applied in a real-life application. Specifically, to employ the proposed difficulty adaptation method, a three-step procedure is proposed.

- The first step is to define the action space $\mathcal{A} = \{a_1, a_2, ..., a_A\}$, which is the design space in the interactive systems. In the visual memory game, the action space is the pre-defined 100 visual memory tasks.
- The second step is to identify the feedback signal g_a. This signal needs to be a quantitative performance measure which can reflect the difficulty level

exerted by the current design choice on the user. In the visual memory game, the memorization time is taken as the difficulty indicator. The longer the memorization time is, the hard the task is for the user. As discussed in Chapter 3, to ensure this quantitation measure is a truthful reflection of difficulty, special care has been taken in the game design.

- The third step is to specify a target value G under this performance measure which the system aims to maintain. The value can be determined by experts or via preliminary study. In the visual memory game, the target memorization time is set to be 4200-5200 ms, which corresponds to Time Bubble No.4. This target memorization range of 4200-5200 ms was selected based on the median memorization time in a preliminary study. The study employed a deterministic adaptation rule with 62 participants: increase (decrease) one target if the answer is correct (wrong). The median memorization time in the result is 4340 ms. The selected target (4200-5200 ms) is the nearest Time Bubble range to the median memorization time.

5.2.2 Experimental Settings

The experiment for obtaining the visual memory profiles via offline training has been covered in the last chapter (see section 4.2 in Chapter 4). This chapter's experiment is focused on the performance for DDA at the online stage. The experiment was conducted using real gameplay data gathered from the Pals online visual memory game platform described in Chapter 3. Three experiment settings were employed: *no adaptation mode*, *rule-based adaptation mode*, and *machine learning adaptation mode*. In all three settings, each player finished 25 tasks. The three modes differ in how the tasks are selected for the players. In the no adaptation setting, the tasks are randomly selected from the question bank throughout the game. This mode serves as a experiment baseline to study whether there are other factors, besides difficulty adaptation, affecting memorization time. In the two adaptive modes, the first 5 tasks were randomly selected, but afterwards, the tasks were selected by the corresponding adaptation algorithms. The goals of the two difficulty adaptation methods are the same, which is to maintain the memorization time at the target level (4200-5200 ms) which is specified in advance. The rule-based adaptation used the incremental adjustment with the target number as difficulty indicator. In particular, if the memorization time is above 5200

ms, the agent will select a task with one more target than the current task and if the memorization time is less than 4200 ms, the agent will select a task with one less target. This is based on the simple assumption that the task difficulty increases with increasing target count. The machine learning-based adaptation used the DDA approach described in Table 4, with the clustering-based personalized difficulty ranking and reinforcement learning-based stochastic adjustment.

The participants were recruited from the Amazon Mechanical Turk platform from Apr-01-2019 to Jun-01-2019. There were 378 players in total. 77 subjects played the no adaptation mode, 136 subjects played the rule-based adaptation mode and 165 subjects played machine learning-based adaptation mode.

5.2.3 Data Analysis and Results

5.2.3.1 Performance Disparity

First, this study examined the research question that was first raised in Chapter 3, which is whether the difficulty adaptation can reduce the performance disparities among the users. The performance disparity is measured by the standard deviation of the memorization times of all the players. A desirable lower performance disparity value implies that the discrepancy between users' performances is moderated. In other words, regardless of the inherent abilities of users, by adapting the difficulty based on user abilities, their performances may be kept at a similar level.

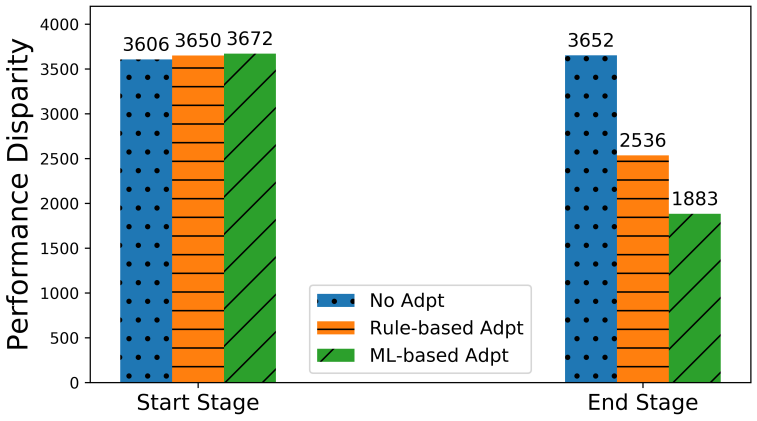


FIGURE 5.8: Performance disparity at the start stage and end stage of the game.

Figure 5.8 plots the performance disparities from the start stage (the first five tasks) to the end stage (the last five tasks) of the game. In the no-adaptation mode, as expected, the performance's standard derivation basically stays the same, with a merely 1.2% of change from 3606 ms to 3652 ms. On the contrary, in the two adaptive modes, there were evident decreases in performance disparity. The rule-based adaptation reduced the performance disparity by 30.5% from 3650 ms to 2536 ms. The machine-learning-based adaptation reduced the performance disparity by 48.7% from 3672 ms to 1883 ms. These results show that the machine-learning based difficulty adaptation has successfully reduced the performance disparity and to a greater extent compared with traditional rule-based difficulty adaptation. As a result, although at the beginning of the game, the performance disparities in the three groups of players were at the similar levels, by the end of the game, the group with the ML-based adaptation achieved much lower performance disparity than the other two (see Figure 5.8).

The heatmap in Figure 5.9 shows a more detailed description of the progressive changes in performance disparity. The performance disparities are presented every five steps. As the game proceeded, the machine learning-based adaptation continued to decrease the performance disparity among players and achieved the lowest level of the three modes evaluated.

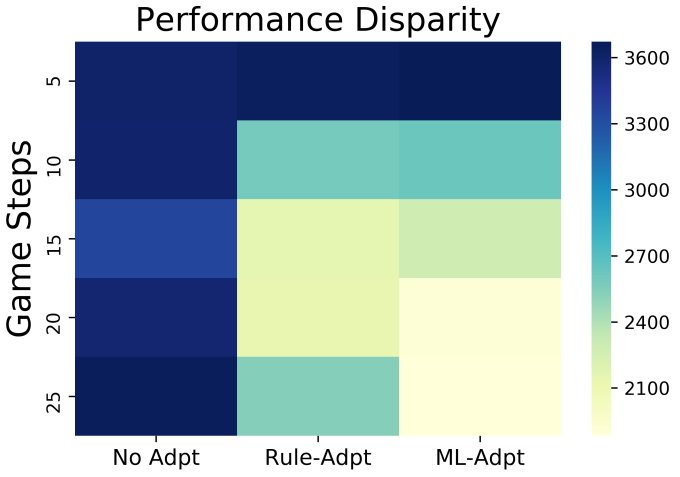


FIGURE 5.9: Heatmap of performance disparity variations as gameplay progresses.

5.2.3.2 Fast and Slow Players

The previous section showed that the proposed machine learning-based adaptation method provides the most effective reduction in performance disparity. This section investigates how the proposed method is able to achieve this better adaptation outcome. Specifically, we examined how the agent adapted differently to faster and slower players.

Note that the first five questions were chosen randomly in all three modes. Thus, the average memorization time for the first five questions was used to infer the memory ability of users. The players were sorted based on this value from lowest to highest. The first 33% were labeled as *fast players* and the last 33% as *slow players*. Their performance changes were examined separately.

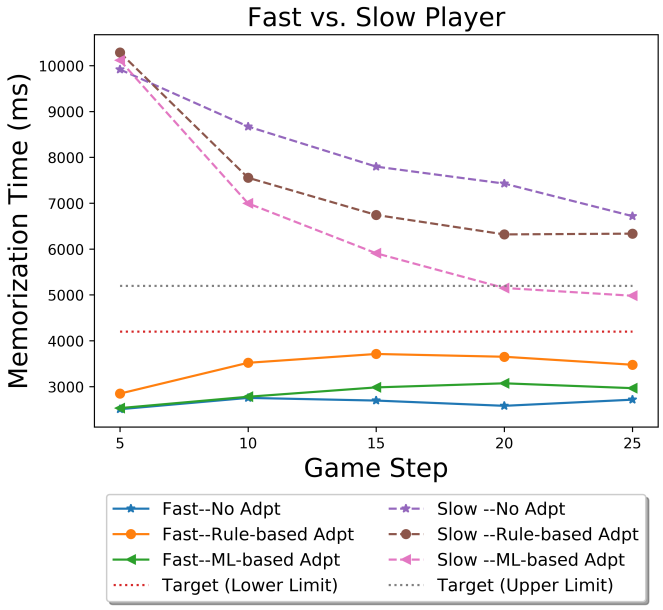


FIGURE 5.10: The memorization time variations in three difficutly adaptation modes for fast and slow players as the game progresses.

As shown in Figure 5.10, the fast players' memorization times were going upward as the game continued. On the other hand, the slow players' memorization times were going downward. This provides an intuitive idea of how performance disparity is reduced, which is through speeding up the slow players and slowing down the fast ones. This result was also investigated from a statistical point of view. The paired t-tests were performed to study whether the memorization time was changed from the start of the game to the end of the game in each mode. For the fast

players, there was no significant change ($t(48) = -0.45$, $p = 0.66$) from the start stage ($M = 2507$, $SD = 613$) to the end stage ($M = 2647$, $SD = 1622$) in the no-adaptation mode, but the memorization time was significantly increased in rule-based adaptation mode ($t(86) = -4.35$, $p < 0.001$, from $M = 2847$, $SD = 870$ to $M = 3563$, $SD = 1344$) and ML-based adaptation mode ($t(106) = -2.64$, $p = 0.011$, from $M = 2530$, $SD = 849$ to $M = 3018$, $SD = 1339$). This result suggests the adaptation can indeed present fast players with more challenging tasks and slow them down. Similarly, the slow players' memorization behaviors were also found to be altered in the two adaptation modes. The memorization time was significantly decreased from the start of the game to the end of the game in rule-based adaptation mode ($t(86) = 7.056$, $p < 0.001$, from $M = 10286$, $SD = 3241$ to $M = 6327$, $SD = 2548$) and ML-based adaptation mode ($t(106) = 11.18$, $p < 0.001$, from $M = 10115$, $SD = 3175$ to $M = 5066$, $SD = 1808$). Interestingly, in the no-adaptation mode, the memorization time of the slow players also significantly declined ($t(48) = 3.08$, $p = 0.006$, from $M = 9917$, $SD = 3130$ to $M = 7072$, $SD = 3915$). Recall that for fast players, no such change was observed in the no adaptation mode. This result implies there is some form of learning effect for the slow players.

This finding raises the question of whether the decline in slow players' memorization time is caused by the learning effect or by adaptation with easier tasks. To answer this question, we directly compared the memorization time in the end stage of the game in the adaptation modes with the no-adaptation mode using independent t-tests. Note there is no significant difference in memorization time at the start stage of the game between the three groups. We investigated whether the difficulty adaptation will make any difference in the memorization time by the end of the game. No significant difference ($t(67) = 0.95$, $p = 0.35$) was observed in end-stage memorization time between the no-adaptation mode ($M = 7072$, $SD = 3915$) and rule-based mode ($M = 6327$, $SD = 2548$). But there is indeed significant difference ($t(77) = 3.08$, $p = 0.003$) between the no-adaptation mode ($M = 7072$, $SD = 3915$) and machine learning-based adaptation mode ($M = 5066$, $SD = 1808$). This result demonstrates the effectiveness of the machine learning-based adaptation in quickly bringing the slow players up to speed, because it led to significant memorization time decline beyond that contributed by learning effects. However, for the rule-based mode, its adaptation effect cannot be differentiated from learning effect and thus there is not enough evidence to indicate that the rule-based adaptation really made any difference for the slow players. In summary, in this experiment,

both adaptation methods successfully adapted to the fast players, but only the machine learning-based method is found to work effectively for the slow players. This is bearing in mind that with the learning effects, the slow players' abilities were improving as the game continued, which brings an extra challenge to the adaptation algorithm. The advantage of the machine learning-based adaptation lies in its ability to quickly detect and accommodate to the user's progressive change in visual memory profile as the users learn and develop new memorization strategies during gameplay. This is achieved with the help of flexible stochastic adjustment and personalized difficulty ranking, which the rule-based adaptation with incremental adjustment and fixed difficulty ranking is unable to do.

To better understand the difference between two adaptation methods, a direct comparison between the two adaptive modes was conducted. The slow players with the machine-learning adaptation ($M = 5066$, $SD = 1808$) have performed significantly faster ($t(96) = 2.83$, $p = 0.006$) than that in the rule-based adaptation mode ($M = 6327$, $SD = 2548$) by the end of the game. But, there is no significant difference ($t(96) = 1.98$, $p = 0.051$) in the fast players' performance between the two modes by the end of the game. This result also indicates the success of machine learning based adaptation lies more in its effectiveness in lifting up the slow players, instead of slowing down the fast ones. Here we seek to understand why it fails to outperform the rule-based method for the fast players. Note that the average memorization time of the fast players at the end of two adaptive modes were both still below the target of 4200-5200 ms (See Figure 5.10). This implies that the question bank probably does not contain the questions that are difficult enough for the fast players. As a result, the adaptation methods do not actually have the opportunity to select the questions that can really challenge the fast players. This probably explains why there is no significant difference in the fast player's memorization time between the two adaptive modes. Hence, for the adaptation method to be effective for both fast and slow players in the participating population, it is important to have a question bank that contains a large number of questions spread over difficulty levels that cover the variations of abilities within the user population.

5.3 Summary

This chapter proposed a stochastic difficulty adjustment method based on reinforcement learning. Unlike traditional heuristic methods, this chapter discussed the fundamental question of how to ensure unbiased convergence of stochastic adjustment and shed some lights on why some heuristic methods fail in certain cases. The proposed method is based on a key idea of bootstrapping policy gradient with better/worse action set to enhance sample efficiency. The idea has the potential to be generalized beyond difficulty adjustment, to other reinforcement learning problems.

The stochastic difficulty adjustment approach proposed in this chapter requires the prior information of difficulty ranking, which can be obtained through the clustering-based personalized difficultly ranking method proposed in Chapter 4. Thus, combining these two techniques leads to a complete algorithm for DDA. The proposed algorithm was applied in the Pals online visual memory game platform described in Chapter 3 and successfully alleviated the "large performance disparity" problem highlighted in Chapter 2 by 48.7% through slowing down the fast players with harder tasks and speeding up the slow players with easier tasks.

Chapter 6

Conclusions

6.1 Discussions

This thesis has made several novel contributions and insightful observations in the areas of HCI and ML.

First, the thesis revealed the existence of the Köhler motivation gain and Köhler discrepancy effects [92] in cognitive-oriented cooperative tasks similar to that observed in physically-oriented ones [94–96]. In the user study with the Stroop game, peer accountability was found to motivate users, especially the weaker partners, to put extra efforts during cooperation. However, it was also observed that large performance discrepancy among the cooperating partners can lead to performance decline. Hence, to maintain the motivation gains, the performance discrepancy between the cooperating partners must be properly moderated. These findings led to a key challenge, namely how we can reduce such performance disparity. A straightforward idea is to present stronger users with harder tasks and weaker users with easier ones. In other words, the challenge levels need to be personalized based on user differences. The traditional interactive systems usually employ a simple difficulty adaptation method of using a scalar value such as the game scores or students' grades to denote users' competence level [18] and presenting harder tasks to the competent users and easier tasks to the less competent ones. These methods suffer from two main drawbacks: oversimplification in user description and rigid adaptation with deterministic or heuristic rules. To overcome these issues in difficulty adaptation, this thesis proposed a machine learning-based solution.

This thesis presented a clustering-based method to identify the personalized difficulty ranking for a given user. To obtain personalized difficulty rankings, this work exploits machine learning's ability in extracting useful information from data and making predictions for unseen cases. Specifically, the proposed algorithm first identifies different types of user difficulty ranking profiles from the pre-collected performance data via clustering and then determines the appropriate user type for a new user in real-time based on NDPM distance. It should be noted that this clustering-based algorithm was developed in particular to address the constraints posed by the interactive systems. First, the training data collected from interactive systems contain a high level of uncertainty arisen from human behaviors. To avoid over-fitting to the noisy training data, instead of using KNN style approaches to directly make inference based on neighboring instances [115], the proposed method employed an extra step of building prototype via clustering and then made a prediction based on the prototype. With this approach, not only is the performance more robust, the run time during the test stage is also reduced. This is because it only compares the new sample with several prototypes instead of all the training instances. Second, a parameter-free method for determination of the cluster number was proposed. Unlike many previous methods, the proposed method is easy to implement and has a low computational cost. And to ensure its robustness, the method was extensively evaluated on various datasets including challenging environments with high dimensions, hierarchical structure or intermix clusters. Lastly, to make a reliable prediction of user type based on a small number of observations during the online stage, the NDMP measure [115, 140, 141] was used to compare the distance based on ranking instead of raw performance measure. As shown in the previous work [115] such measure can work robustly with some dimensional values missing and thus can support prediction with limited play history. It is these design considerations that give the proposed clustering-based method the ability to support responsive interactions in a timely and robust manner. The results in the visual memory game example showed that the clustering-method can make an accurate prediction of user visual memory profiles with minimal number of gameplay observations. The identified user profiles were able to embody complex user differences, such as memorizing strategies and visual memory characteristics (e.g. the ability to detection structure patterns). In addition, while the clustering-based method was applied in a visual memory game, it may also be generalized to many other contexts. In fact, the complex and user-variable difficulty characteristics

of the visual memory game are representative of typical task difficulty encountered in many real-life scenarios. For example, in the online education systems like MOOCs, depending on their previous education backgrounds, students can exhibit substantial differences in their difficulty ranking profiles. The proposed difficulty ranking personalization method can be used to inform the teachers which areas the students need to strengthen and help the teachers manually tailor the practicing question sets for the students. Moreover, since the method is computationally efficient, it can be used in real-time interactive systems to guide automatic difficulty adaptation.

This thesis also proposed a stochastic difficultly adjustment method with theoretical convergence guarantee. Formalizing difficulty adjustment problem in reinforcement learning, a novel framework was proposed to improve the sample efficiency of policy gradient making it applicable in real-time responsive systems. The proposed BPG framework was able to accomplish three goals. First, by introducing the concept of better/worse action, BPG incorporates the prior information of personalized difficulty ranking into the adaptation mechanism, achieving personalized adjustment decisions for each user. Second, by updating the probability of better/worse actions, the batch size required in policy gradient ends up being highly reduced, overcoming the sample efficiency in human-in-the-loop applications. More importantly, a theorem was provided to ensure the stochastic adjustment converge. Interestingly, this theorem not only proved the unbiased convergence of the proposed adjustment mechanism but also shed lights on why some previous heuristic adjustment rules [23], despite seeming reasonable, often fail in practice. With this bootstrapping technique, the proposed adjustment mechanism achieved fast and unbiased convergence with a small batch size both theoretically and empirically.

This work has also led to a number of insight observations regarding several classes of machine learning algorithms. In the context of cluster analysis, this thesis showed that the curvature of the cost function, in terms of the within-cluster variance, actually contains valuable clues about the number of clusters existed in the data. These hidden clues which have been heuristically explored via visual inspection [123, 126], have never been seriously treated or systematically studied in previous works. This research made a key finding regarding why it is hard to extract useful cluster information from the cost function curve: there is an irrelevant element obscuring the curvature information, which is the scaling factor in axes of the cost function

plot. After applying theoretical analysis to eliminate the influence of the scaling factor, it turns out that the knowledge of cluster number can be revealed from the curvature in a surprisingly easy way. The computational cost is substantially lower than the previous cluster number determination methods [125, 130]. In the context reinforcement learning, the proposed surrogate policy gradient theorem not only answered how to achieve unbiased convergence while using BPG framework, but, more importantly, opened up a new research direction in enhancing the sample efficiency of policy gradient. As we know, policy gradient is very popular in RL due to its stable convergence characteristic, but suffers from the high variance problem. Thus RL researchers have endeavored to devise surrogate gradients with a lower variance to replace the original policy gradient method. It was previously believed that to maintain stable convergence, special care needs to be taken in the design of new surrogate gradient to make it equal to the original policy gradient [32, 33, 154, 155, 159]. However, this research showed that this constraint is actually unnecessary. As shown in the theorem 5.1.1, sometimes it does not matter if the surrogate gradient does not equal policy gradient. As long as the proposed sufficient condition is met, the surrogate gradient can still achieve unbiased convergence to the optimal solution. With this key constraint lifted, this research pointed out a new class of surrogate gradients which have the potential to reduce variance more aggressively at the same time maintaining stable convergence.

6.2 Generalizations and Limitations

Combining the clustering-based difficulty ranking personalization and RL-based adjustment, the overall ML-based adaptation approach achieved effective personalizing of task difficulty in the visual memory game platform. While the proposed DDA method was applied in an environment involving only simple cognitive activities, it can potentially be applied to solve the challenges faced by many other real-life applications. For example, the educational applications are often faced with the challenge of a large question bank with hundreds or even thousands of question candidates [12, 103, 115] and a diverse user base with different backgrounds and abilities. The proposed method brings an opportunity to detect the different types of students and to explore large action space efficiently. However, there are some cases in which the proposed method is not applicable due to some

practical limitations. This section discusses the generalizations and limitations of the proposed method in other application contexts. Recall that applying the ML-based DDA method in visual memory game involves a three-step procedure as described in Chapter 5. This three-step process can be used to analyze whether the method is suitable for an application and if so, how it can be applied.

- The first step is to define the action space $\mathcal{A}$, i.e. the design space. This is straightforward for some applications. For example, in the visual memory game, each visual memory task is taken as an action. But other applications may involve continuous design spaces or multi-dimensional spaces. As the proposed method deals only with a discrete design space, multi-dimensional continuous spaces need to be transformed into discrete ones via discretization and coding. Take a shooting game as an example. The choices of enemy settings can be considered as actions. The enemy settings can consist of multiple dimensions such as weapon types and position configurations. Some dimensions can involve continuous variables, e.g. positions. In this case, the continuous space of the position needs to be discretized to discrete choices. Then the multi-dimension spaces of weapon type and position need to be coded together to a one-dimensional discrete space of enemy settings, such as setting No.1=[weapon No.1, position No.1], setting No.2=[weapon No.1, position No. 2], etc.

- The second step is to obtain a feedback signal g_a for difficulty. Possible signals include the graded response in education systems, the time taken in completing the task, or the score received in games. In the shooting game example, the in-game performance like the damage of the player under an enemy setting can be used to indicate the challenge level of that enemy setting. The key point in designing such difficulty signal is to ensure the signal is reliable and gives a quantitative measure of difficulty. To this end, sometimes special care needs to be taken in the game design. For instance, in the visual memory game, the scoring mechanism penalizes the inaccurate memorization attempts as well as slow memorizing actions, aiming to discourage the reckless guessing behavior and encourage focused best-effort memorization behaviors. Assuming players are motivated to achieve the best score,

this stimuli design serves to reduce undesirable noise in the relationship between the quantitative performance measure (memorization time) and task difficulty.

- Thirdly, a target value under this performance measure needs to be specified in advance. The proposed method only answers the question of how to maintain a given target performance. How to select a target value is beyond the scope of the proposed method. The target value can be determined by experts or via preliminary studies. Based on the aim of the systems, the target value can vary. For instance, in a computerized adaptive testing system, the target performance is set to be 0.5 in order to maximize the information gain, but in an educational environment, the target performance can be set a bit higher (e.g 0.75) so as to promote the learning process [5].

Following this three-step procedure, the proposed method can be applied in the shooting game example by taking the choices of enemy settings as actions and the damage as difficulty measure and specifying a target damage level. The clustering-based method can first be applied to examine if there is a universal difficulty ranking for different users, in terms of enemy settings. After understanding which enemy settings are considered challenging or easy for the player, the BPG-based adjustment method can be applied to select enemy settings to maintain the target player performance. However, it should be noted in some systems, one may have difficulty in performing these three steps. For example, regarding the first step of design space, some applications may involve a high-dimensional domain, such as a maze game in which the challenge level is controlled by the positions of many obstacles. In this case, the discretization is inefficient and may lead to a prohibitively large number of actions. When it comes to the second step, reliable quantitative difficulty measures are perhaps not available in some applications. For example, in an intelligent tutoring system with multiple-choice questions, the grade for a question is only binary (i.e. correct/incorrect) rather than numeric. As a result, the difficulty is only indicated at a coarse level. Moreover, it is hard to determine a target value under this binary performance measure. The proposed difficulty adaptation method is thus not directly applicable in this case. To apply the proposed method, players probably need to play several questions from a question pool consisting of tasks with a similar difficulty level and then the error rate computed from the combination of several consecutive answers can provide

a meaningful quantitative difficulty measure. For instance, if a player answered 4 questions from the same difficulty level and only 1 of them were answered correctly, the error rate in this case is 0.75. Given a specified target error rate (e.g. 0.5) [5, 103], this feedback signal suggests the current questions are too hard for the player. The proposed BPG adjustment mechanism can be applied in this case to increase the probabilities of easier questions and decrease the probabilities of harder ones in order to select questions suitable challenging for the player. Finally, with respect to the third step, it is likely that some application goals cannot simply be summarized by a feedback signal. For instance, in a system to maximize pre-test/post-test learning outcomes, the final outcome only comes at the end of the game. There is no feedback signal to guide the difficulty adjustment along the way. In these scenarios, the proposed method cannot be applied. Next section of future work will revisit these limitations and discuss how to extend the method to wider application scenarios.

6.3 Future Research Directions

There are some limitations in this work that offer great opportunities for further exploration.

First, as mentioned before this work is mainly concerned with the adaptation of challenge levels in a discrete design space (e.g. Task No.1,..., Task No.100). In some other applications, the challenge levels can be controlled in a (multi-dimension) continuous design space. For example, in the flappy bird game, the difficulty can be tuned by changing the horizontal spacing and vertical gap between pipes [7]. Apart from discretization which may lead to a large number of actions, a more promising solution is to use the proposed continuous BPG. Continuous BPG can directly explore muli-dimensional continuous space. Its efficiency, especially in high dimensional cases (e.g. 60 dim) has been demonstrated in a simulated environment in Chapter 5. Further experiments of continuous BPG need to be conducted in real-world applications to validate its effectiveness.

So far, this thesis has focused on one particular kind of personalization - personalized difficulty level. Other kinds of personalization in the interactive systems, like selecting display choices [21, 163] or command set [20], can also be formalized in

the framework of reinforcement learning framework, in which the proposed BPG method can also be employed. However, one should be mindful of a few considerations while applying the proposed method in these applications. First, it should be noted that the value proposition of the BPG method lies in its efficiency in exploring large action space. With only a few action candidates like choosing from three display message or seven education concepts [21, 163], the advantage of BPG over other traditional RL algorithms may not be very significant. Second, these applications with other personalization goals may not care about the difficulty ranking but other aspects of users' differences. In these cases, the clustering-based difficulty ranking method is not applicable. A more general approach is to incorporate the detection process into the reinforcement learning framework with actor-critic methods. The critic function can then be learned to denote the desirability of the design options in both continuous and discrete design space. A simple example of this actor-critic method has been shown in the simulation experiment with continuous BPG, in which the critic function of $Q(a)$ is learned via regression. To achieve personalization, the MDP framework can be employed to replace $Q(a)$ with $Q(s, a)$. In this way, the state space s can be used to capture user information. In fact, extending BPG to MDP framework has another important advantage. It can capture the long-term effects of design decisions. Note that the current BPG method requires timely feedback signals. But some applications may contain delayed feedback. As discussed earlier, in an educational application seeking to improve pre-test/post-test learning outcomes, the feedback signal does not come until the very end. In this case, BPG is not applicable, but the MDP framework can take the long-term effects of actions into account by maximizing the cumulative reward instead of the immediate reward. Hence, extending BPG from MAB to MDP is another research direction that deserves further investigation.

Lastly, this research on difficulty adaptation was originally motivated by the observed negative influence of "large performance disparity" issue in the cooperative Stroop game described in Chapter 2. A natural future work is to revisit this issue by employing the proposed difficulty adaptation mechanism to reduce the discrepancy of the partners in the Stroop game. Note that there is no obvious difficulty variation among the different Stroop tasks as one color selection is not significantly harder than another. We need therefore change the concept of difficulty adaptation from one of selecting an appropriate task to match user's ability to one of giving the appropriate advantage or disadvantage to normalized the user's ability. In the case

of the cooperative Stroop game, the stronger players could be given a longer delay before the task is presented to them so that slower players have a higher chance to complete their task at about the same time. In other words, the action space for the difficulty adaptation is no longer the task candidates but the value of the delay in the task presentation. Following the three-step procedure, this continuous delay time can be discretized to discrete space, like $[0s, 0.01s, ..., 0.09s, 0.1s]$. The performance discrepancy of the two partners can be considered as the feedback signal. And a target discrepancy level can be specified, e.g. zero or a small number. Then the proposed adaptation mechanism can be employed to stochastically adjust the delay value to keep the performance discrepancy at the target level.

6.4 Conclusions

This thesis has taken a step towards exploiting the intelligence of machine learning to improve human-computer interaction. With the novel techniques to overcome the practical constraints posed by interactive systems, machine learning algorithms were successfully applied in real-time responsive applications to personalize difficulty adaptation for individual users. We now have the ability to incorporate human-in-the-loop ML algorithms in creating personalized interactive systems design that can potentially be more effective in maintain the appropriate state of flow for each and every user.

Appendix A

Supplementary Material for Visual Memory Profile

A.1 Difficulty Ranking Personalization with Different Amounts of Training Data

In Chapter 4, the proposed difficulty ranking personalization (DRP) method was conducted with a training data of 544 players. This section shows the results of DRP with other amounts of training data. Figure A.1 plots the prediction quality of difficulty ranking at different game stages with clustering on the different numbers of training samples. As the number of training samples increases, the NDMP decreases in general. This suggests the DRP seem to achieve a more accurate prediction of difficulty ranking with more training data. However, this statement is not true for predictions at the early stages of the game, i.e. with 5 steps and 10 steps of gameplay records. With these small numbers of observations, the prediction performance declines as the training samples increase from 544 samples to 760 samples. This result implies the prediction at the early stage of the game can deteriorate with too many training samples.

To further investigate this phenomenon, the clustering results, in terms of the cluster number is shown in Table A.1. The Curvature-based method identified 3 clusters with 544 samples and 7 clusters with 760 samples. As we know, a relatively large number of clusters can lead to over-fitting to the training data and reduce

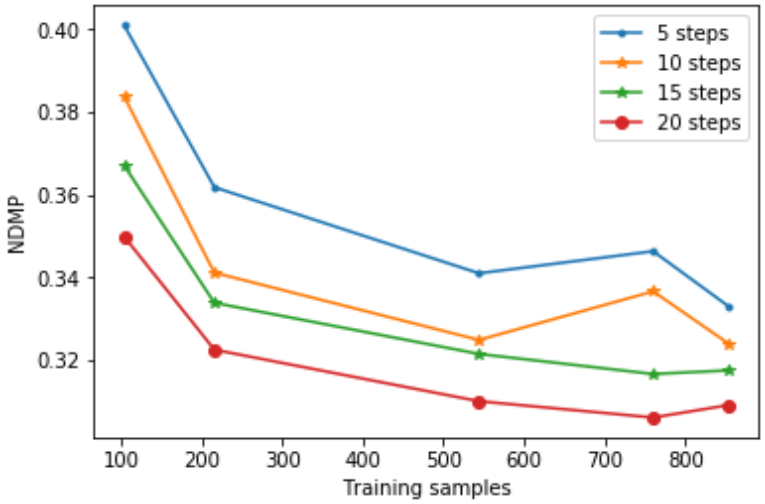


FIGURE A.1: Prediction error of difficulty rankings at the different game stages with different amounts of training samples.

the generalization ability. This may explain the decline in performance with 760 samples (7 clusters). In fact, the prediction of difficulty ranking at early game stage is quite important for difficulty adaptation as it directly impacts the subsequent adaption behavior. Therefore, the clustering result with 3 clusters is chosen as the predicted difficulty ranking profiles and is used for the subsequent difficulty adaptation study.

Number of Training samples	Predicted Cluster Number
104	8
216	4
544	3
760	7
853	6

TABLE A.1: Training batch and number of clusters.

A.2 Question Bank

The 100 randomly generated questions used in the visual memory game are shown in Figure A.2. Specifically, when generating the tasks, the task number is specified and the positions of the targets are generated randomly. The target number lies in between 3 to 8.

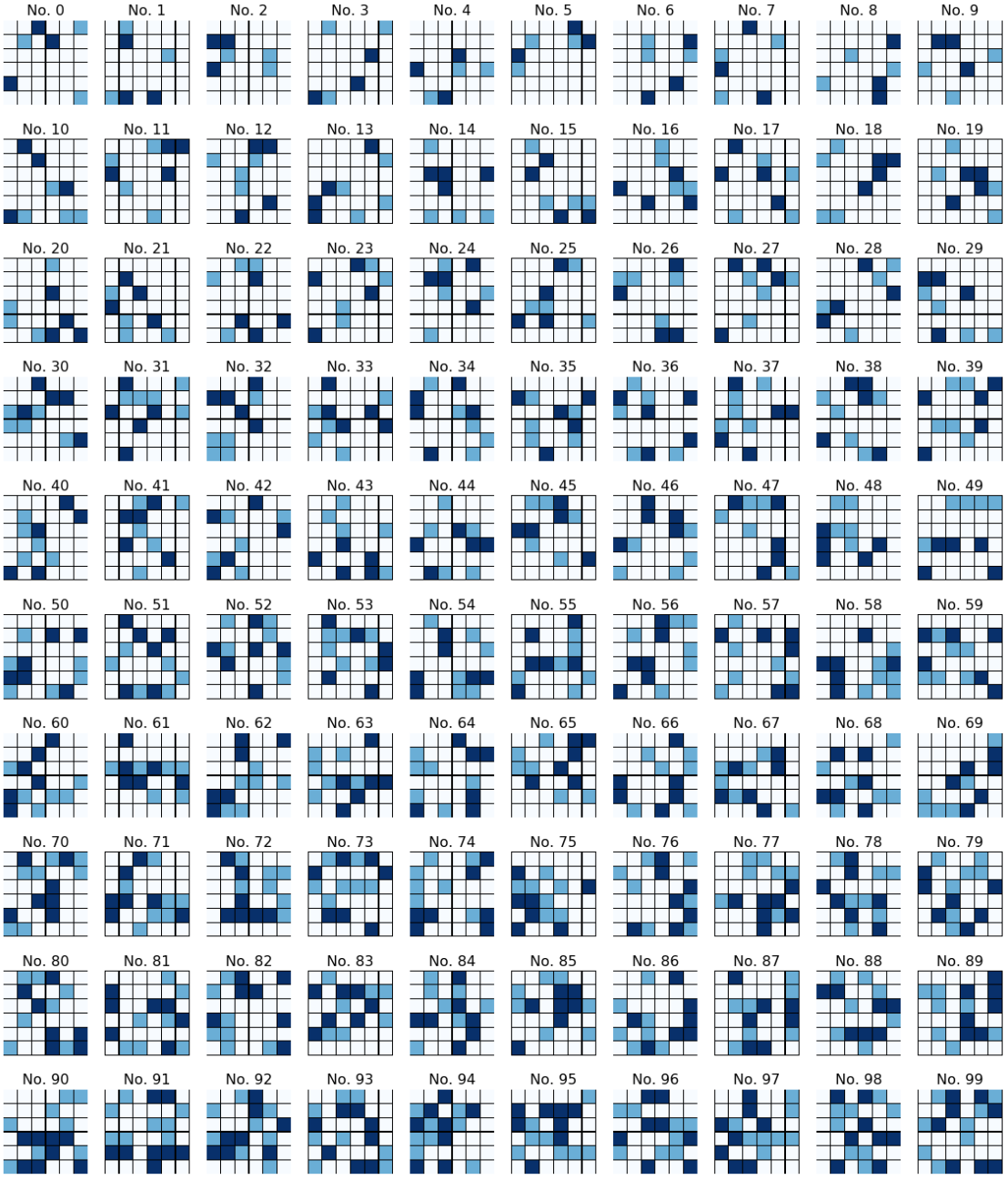


FIGURE A.2: Question bank for the visual memory game.

Appendix B

Supplementary Material for Clustering

B.1 Six Baseline Approaches for Determination of Cluster Number

The CH [120] method, in short, chooses the number of clusters as the argument maximizing eq. B.1 where $J(k)$ is within-cluster variance with k clusters and n is the number of observations.

$$CH(k) = \frac{(J(1) - J(k))/(k-1)}{J(k)/(n-k)} \tag{B.1}$$

The approach of Krzanowski and Lai [122] maximizes $KL(k)$ given by eq. B.2:

$$\begin{aligned} KL(k) &= \left| \frac{DIFF(k)}{DIFF(k+1)} \right| \\ DIFF(k) &= (k-1)^{2/p} J(k-1) - K^{2/p} J(k), \end{aligned} \tag{B.2}$$

where p is a dimension of the data which is used for adjustment of $DIFF(k)$.

Hartigan et al. [121] proposed choosing the smallest value of k such that $H(k) \leq 10$ in eq. B.3. $H(k)$ is effectively a partial F statistic for testing whether it is worth

adding a $(k+1)$st cluster to the model:

$$H(k) = (n-k-1)[\frac{J(k)}{J(k+1)} - 1]. \tag{B.3}$$

Kaufman and Rousseeuw [130] proposed silhouette width as shown in eq. B.4, measuring how well the ith point is clustered. The term $a(i)$ is the average distance between the ith point and all other observations in its cluster, and $b(i)$ is the average distance to points in the nearest cluster, where nearest is defined as the cluster minimizing $b(i)$. The number of clusters that maximizes the average value of $s(i)$ is chosen:

$$s(i) = \frac{b(i) - a(i)}{\max[a(i), b(i)]}. \tag{B.4}$$

The Gap approach developed by Tibshirani et al. [125] is described in eq. B.5:

$$Gap(k) = {}^{1}/_{B} \sum_{b} \log(J_b^*(k)) - \log(J(k)). \tag{B.5}$$

In this method, B different uniform datasets, each with the same range as the original data, are produced, and the within-cluster variance is calculated for different numbers of clusters. $J_b^*(k)$ is the within-cluster variance for the bth uniform dataset. To avoid adding unnecessary clusters, an estimate S_k of the standard deviation of $\log(W_b^*(k))$ is produced, and the smallest value of k is chosen as the number of clusters, such that

$$Gap(k) \geq Gap(k+1) - S_{k+1}.$$

Finally, the Jump method proposed by Sugar et al. [124] maximizes the $Jump(k)$ given in eq. B.6.

$$Jump(k) = J(k)^{-p/2} - J(k-1)^{-p/2}. \tag{B.6}$$

The transformation parameter p is typically chosen as half of the space dimension.

B.2 Experimental Settings of Synthetic Datasets

The following 5 data arrangements (simulations) was used to test the Curvature-based method:

- Five basic Gaussian clusters in 2 dimensions. This simulation is designed to test the performance on basic Gaussian clusters. One cluster is placed in the middle, and four other clusters are spaced with a separation of 2.5 from the center cluster in each dimension (see Figure 4.5(a)).
- Five elongated clusters in 2 dimensions. This simulation is aimed at investigating the performance of the methods when there was some dependence among the dimensions. Specifically, there is a correlation of 0.7 between the dimensions. The placement of clusters is the same as in the previous case (see Figure 4.5(b)).
- Five clusters with different shapes in 2 dimensions. This simulation is designed to test the effect of differing correlation. The correlations for 5 clusters are 0.0, 0.7, 0.3, 0.3 and 0.7. The placement of clusters is the same as in the two previous cases (see Figure 4.5(c)).
- Five Gaussian clusters in 10 dimensions. In this simulation, the performance of approaches on highly multivariate data is examined. A basic 10-dimensional mixture of five Gaussian clusters is generated. The five clusters are spaced on a line with a separation of 1.6 in each dimension.
- The last simulation is used to compare the methods on non-Gaussian data. We generated 4 clusters in 2 dimensions using exponential distributions with mean 1 independently in each dimension. The clusters were arranged on a square with sides of length 4 (see Figure 4.5(d)).

All the above-described data arrangements have standard deviations of 1 in each dimension. In simulations $1-4$, the distances between the centers of the middle clusters and the centers of surrounding clusters are equal to 2.5, 3.0, 2.5 and 1.6, respectively, in each dimension. Each simulation set consists of 400 observations equally divided among clusters.

Initially, for each simulation, 50 datasets were randomly generated. Next, for each dataset we ran k-Means algorithm with 20 restarts and then applied the Curvature method and the 6 other methods to estimate the optimal number of clusters.

B.3 Clustering Results on the Real-World Datasets

TABLE B.1: Experimental comparison (first and second candidates) of Curvature method with six other approaches on 20 real-world datasets. A star (*) sign denotes the correct results; a plus (+) sign denotes the data sets, which have two reasonable (alternative) cluster numbers.

Dataset	True number	CH		KL		Hartigan		Silhouette		Gap		Jump		Curvature	
		1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd
Spectf[164]	2	2*	3	2*	5	6	7	2*	3	9	5	10	9	2*	3
Ionosphere[2]	2	2*	3	2*	8	8	9	2*	9	9	5	10	9	2*	4
Breast cancer[165]	2	8	9	2*	8	10	8	2*	9	7	5	10	8	2*	3
Parkinsons[166]	2	9	10	4	4	9	10	3	3	8	4	10	9	3	2*
Haberman[167]	2	4	2*	4	4	10	9	2*	4	2*	1	7	4	4	2*
Transfusion[168]	2	9	10	9	7	8	10	2*	3	1	4	9	8	8	9
Magic[169]	2	2*	3	5	5	9	10	2*	3	9	5	9	8	2*	5
Musk[170]	2	3	2*	3	9	8	6	3	6	9	5	9	8	3	10
MiniBooNE[138]	2	2*	3	2*	6	9	10	/	/	/	/	2*	6	2*	3
Skin[139]	2	2*	5	2*	7	8	7	/	/	/	/	10	4	2*	4
Hill[171]	2	8	9	2*	3	10	8	2*	3	6	5	3	7	2*	3
Seed[1]	3	3*	2	3*	2	10	9	2	3*	3*	4	8	9	2	3*
Cardiotocography[172]	3,10^{+}	3*	7	2	3*	8	5	2	3*	9	4	10*	9	10*	2
Wine[135]	3	10	9	2	7	8	9	2	3*	1	2	10	9	3*	2
Iris[173]	3	3*	4	2	8	8	10	2	3*	9	5	3*	2	2	3*
Sensor[135]	4	2	3	2	4*	10	9	2	3	9	5	10	9	3	4*
Breast tissue[3]	4,6^{+}	10	9	4*	2	10	9	1	9	4*	5	10	9	4*	2
Vehicle[135]	4	2	6	2	6	9	10	2	3	9	5	10	10	9	2
Winequalityred[174]	6	10	7	2	3	9	7	2	3	1	6*	10	9	7	2
Statlog land[135]	6	3	4	3	4	9	7	2	3	9	5	10	9	3	6*

Appendix C

Proof of Theorems

C.1 Proof of Theorem 5.1.1

Proof. **Proof of a_* is the optimal solution** The gradient regarding each softmax weight is: $\tilde{\nabla}_{w_k} J(\theta) = \pi_\theta(a_k)(f_\theta(a_k) - E_{a\sim\pi_\theta}[f_a])$. Let $\tilde{\nabla}_{w_k} J(\theta) = 0, \forall k = 1..A$, we have:

$$\pi_\theta(a_k) = 0 \text{ or } f_\theta(a_k) - E_{a_j\sim\pi_\theta}[f_\theta(a_j)] = 0, \forall k = 1, ..., A. \tag{C.1}$$

Based on Proposition C.2, which will be stated shortly, when initialized with $\pi_\theta(a_j) = \frac{1}{A}, j = 1, ..., A$, the optimal action a_* has the highest probability during all the gradient ascent iteration steps, i.e. $\pi_\theta(a_*) \geq \pi_\theta(a)$.. Since the sum of all the action probability should be 1, we have $\pi_\theta(a_*) > 0$. Together with Eq (C.1), we have

$$\begin{aligned} 0 &= f_\theta(a_*) - \mathbb{E}_{a_j\sim\pi_\theta}[f_\theta(a_j)] \\ &= \sum_{a_j \neq a_*} \pi_\theta(a_j)[f_\theta(a_*) - f_\theta(a_j)]. \end{aligned} \tag{C.2}$$

Since $\forall a_j \neq a_*, f_\theta(a_*) \geq f_\theta(a_j)$, to satisfy the above equation, we have:

$$\pi_\theta(a_j)[f_\theta(a_*) - f_\theta(a_j)] = 0, \forall a_j \neq a_*. \tag{C.3}$$

Based on the second condition on $f_\theta(a)$, we have: $\forall a_j \neq a*$, if $0 < \pi_\theta(a_j) < 1$ and $\pi_\theta(a_*) \neq 0$, then $f_\theta(a_*) > f_\theta(a)$. Thus, to satisfy Eq (C.3), we have $\pi_\theta(a_j) = 1$ or $\pi_\theta(a_j) = 0, \forall a_j \neq a*$. From the Proposition C.2 ($\pi_\theta(a_*) \geq \pi_\theta(a_j)$) and the fact the sum of all the action probability should be 1, we have $\forall a_j \neq a_*, \pi_\theta(a_j) \neq 1$. Thus,

to satisfy the Eq (C.3), we have: $\pi_\theta(a_j) = 0, a_j \neq a_*$.

Proof of convergence To prove that the policy optimization will converge to a_*, we prove that at each iteration step: $\pi_\theta^{t+1}(a_*) > \pi_\theta^t(a_*)$ if $\exists a_{k0} \neq a_*, \pi_\theta(a_{k0}) \neq 0$. From the definition of softmax policy, we have: $\pi_\theta(a_*) = \frac{e^{w_*}}{e^{w_*} + \sum_{a_k \neq a_*} e^{w_k}} = \frac{1}{1 + \sum_{a_k \neq a_*} e^{w_k - w_*}}$. Therefore, to prove $\pi_\theta^{t+1}(a_*) > \pi_\theta^t(a_*)$, we just need to prove $\Sigma_{a_k \neq a_*} e^{w_k - w_*}$ decreases from step t to step $t+1$. We have: $\forall a_k \neq a_*$, $(w_*^{t+1} - w_k^{t+1}) - (w_*^t - w_k^t) = \alpha \nabla_{w_*^t} J - \alpha \nabla_{w_k^t} J = \alpha \pi_{\theta^t}(a_*) \cdot (f_{\theta^t}(a_*) - E_{a \sim \pi_{\theta^t}}[f_{\theta^t}(a)]) - \alpha \pi_{\theta^t}(a_k) \cdot (f_{\theta^t}(a_k) - E_{a \sim \pi_{\theta^t}}[f_{\theta^t}(a)])$.

Based on the first condition of $f_\theta(a)$, i.e. $f_\theta(a_*) \geq f_\theta(a), \forall \theta$, we have $(f_{\theta^t}(a_*) - E_{a \sim \pi_{\theta^t}}[f_{\theta^t}(a)]) \geq (f_{\theta^t}(a_k) - E_{a \sim \pi_{\theta^{t-1}}}[f_{\theta^t}(a)])$ and $(f_{\theta^t}(a_*) - E_{a \sim \pi_{\theta^t}}[f_{\theta^t}(a)]) \geq 0$. From Proposition C.2, we have that during all the iteration step $\pi_{\theta^t}(a_*) \geq \pi_{\theta^t}(a_k)$. Therefore we have

$$w_*^{t+1} - w_k^{t+1} \geq w_*^t - w_k^t, \forall a_k \neq a_*. \tag{C.4}$$

Moreover, based on $\pi_{\theta^t}(a_*) \geq \pi_{\theta^t}(a)$, we have $\pi_{\theta^t}(a_*) \neq 0$ and $\pi_\theta(a_{k0}) < 1$. Together with $\pi_\theta(a_{k0}) \neq 0$, and the second condition of $f_\theta(a)$, we have that $f_\theta(a_*) > f_\theta(a_{k0})$. Hence, $w_*^{t+1} - w_{k0}^{t+1} > w_*^t - w_{k0}^t$. Combined this with Eq (C.4), we show $\Sigma_{a_k \neq a_*} e^{w_k - w_*}$ decreases from t to $t+1$. In other words, $\pi_\theta^t(a_*)$ will always increase until all the non-optimal actions have zero probability. □

C.2 Proof of Proposition C.2

Given a surrogate policy gradient defined as $\tilde{\nabla}_\theta J(\theta) = \Sigma_{a_k} f_\theta(a_k) \nabla_\theta \pi_\theta(a_k)$, *where* $a_k \in \mathcal{A} = \{a_1, .., a_A\}$ *and* $\pi_\theta(a_k) = \frac{e^{w_k(\theta)}}{\Sigma_i e^{w_i(\theta)}}$ *is a softmax exploration policy parameterized by* θ *and is initialized with* $\pi_\theta(a_k) = \frac{1}{A}, j = 1, ..., A$. *If there exists an action, which has the highest value of* $f_\theta(a)$ *for any* θ, *i.e.* $\exists a_*, \forall \theta, f_\theta(a_*) \geq f_\theta(a)$, *then during the all the gradient ascent iteration steps,* a_* *always has the highest exploration probability, i.e.* $\pi_{\theta^t}(a_*) \geq \pi_{\theta^t}(a), \forall \theta$.

Proof. At the first step, we have $\pi_{\theta^{t=1}}(a_*) = \pi_{\theta^{t=1}}(a_k)$. And it is easy to show that at the following steps $t > 1$, if $\pi_{\theta^{t-1}}(a_*) \geq \pi_{\theta^t}(a_k)$, then $\pi_{\theta^t}(a_*) \geq \pi_{\theta^t}(a_k)$ as

follows:

$$\begin{aligned} w_*^t &= w_*^{t-1} + \alpha\pi_{\theta^{t-1}}(a_*) \cdot (f_{\theta^{t-1}}(a_*) - E_{a\sim\pi_{\theta^{t-1}}}[f_{\theta^{t-1}}(a)]) \\ &\geq w_k^{t-1} + \alpha\pi_{\theta^{t-1}}(a_k) \cdot (f_{\theta^{t-1}}(a_k) - E_{a\sim\pi_{\theta^{t-1}}}[f_{\theta^{t-1}}(a)]) \\ &= w_k^t, \forall k. \end{aligned}$$

The two equalities use the definition of gradient. The inequality, where the main work happens, uses the property of $f(a_*)$: $\forall\theta, f(a_*) \geq f(a)$ and the condition $\pi_{\theta^{t-1}}(a_*) \geq \pi_{\theta^t}(a_k)$. Therefore, we have $\forall t, \pi_{\theta^t}(a_*) \geq \pi_{\theta^t}(a_k)$. □

List of Author's Awards, Patents, and Publications[1]

Award

- **Student Travel Award**, *AAMAS 2019* (International Conference on Autonomous Agents and Multiagent Systems).

Journal Articles

- **Yaqian Zhang**, Jacek Mańdziuk, Chai Hiok Quek, Wooi-Boon Goh. Curvature-based method for determining the number of clusters. *Information Sciences*, 415:414-428, 2017.
- **Yaqian Zhang** and Wooi-Boon Goh. The influence of peer accountability on attention during gameplay. *Computers in Human Behavior*, 84:18-28, 2018.

Conference Proceedings

- **Yaqian Zhang** and Wooi-Boon Goh. Bootstrapped policy gradient for difficulty adaptation in intelligent tutoring systems. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 711-719. International Foundation for Autonomous Agents and Multiagent Systems, 2019.

[1]The superscript * indicates joint first authors

Bibliography

[1] Małgorzata Charytanowicz, Jerzy Niewczas, Piotr Kulczycki, Piotr A Kowalski, Szymon Łukasik, and Sławomir Żak. Complete gradient clustering algorithm for features analysis of x-ray images. In *Information technologies in biomedicine*, pages 15–24. Springer, 2010. xviii, 59, 136

[2] Vincent G Sigillito, Simon P Wing, Larrie V Hutton, and Kile B Baker. Classification of radar returns from the ionosphere using neural networks. *Johns Hopkins APL Technical Digest*, 10(3):262–266, 1989. xviii, 61, 136

[3] J Jossinet. Variability of impedivity in normal and pathological breast tissue. *Medical and Biological Engineering and Computing*, 34(5):346–350, 1996. xviii, 61, 136

[4] Brian Christian. The a/b test: Inside the technology thats changing the rules of business. *Wired Magazine*, 20(5), 2012. 1

[5] Jan Papoušek, Vít Stanislav, and Radek Pelánek. Impact of question difficulty on engagement and learning. In *International Conference on Intelligent Tutoring Systems*, pages 267–272. Springer, 2016. 1, 2, 3, 124, 125

[6] Alexander Zook, Eric Fruchter, and Mark O Riedl. Automatic playtesting for game parameter tuning via active learning. In *FDG*, 2014. 1

[7] Mohammad M Khajah, Brett D Roads, Robert V Lindsey, Yun-En Liu, and Michael C Mozer. Designing engaging games using bayesian optimization. In *Proceedings of the 2016 chi conference on human factors in computing systems*, pages 5571–5582. ACM, 2016. 2, 125

[8] Anna Raffert, Matei Zaharia, and Thomas Griffiths. Optimally designing games for cognitive science research. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 34, 2012. 1

[9] Sandra Sampayo-Vargas, Chris J Cope, Zhen He, and Graeme J Byrne. The effectiveness of adaptive difficulty adjustments on students' motivation and learning in an educational computer game. *Computers & Education*, 69: 452–462, 2013. 2, 3, 39, 40, 87

[10] Jelena Nakic, Andrina Granic, and Vlado Glavinic. Anatomy of student models in adaptive learning systems: A systematic literature review of individual differences from 2001 to 2013. *Journal of Educational Computing Research*, 51(4):459–489, 2015.

[11] Travis Mandel, Yun-En Liu, Sergey Levine, Emma Brunskill, and Zoran Popovic. Offline policy evaluation across representations with applications to educational games. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 1077–1084. International Foundation for Autonomous Agents and Multiagent Systems, 2014. 90

[12] Andrew S Lan and Richard G Baraniuk. A contextual bandits framework for personalized learning action selection. In *EDM*, pages 424–429, 2016. 2, 90, 91, 122

[13] Michael J Pazzani and Daniel Billsus. Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer, 2007. 2

[14] Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. Personalized news recommendation based on click behavior. In *Proceedings of the 15th international conference on Intelligent user interfaces*, pages 31–40. ACM, 2010. 2

[15] Changchun Liu, Pramila Agrawal, Nilanjan Sarkar, and Shuo Chen. Dynamic difficulty adjustment in computer games through real-time anxiety-based affective feedback. *International Journal of Human-Computer Interaction*, 25 (6):506–529, 2009. 2, 39, 40, 87

[16] Michael Booth. The ai systems of left 4 dead. In *Artificial Intelligence and Interactive Digital Entertainment Conference at Stanford, 2009*, 2009. 3

[17] Robin Hunicke. The case for dynamic difficulty adjustment in games. In *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology*, pages 429–433. ACM, 2005. 2, 39, 40

[18] Gustavo Andrade, Geber Ramalho, Hugo Santana, and Vincent Corruble. Challenge-sensitive action selection: an application to game balancing. In *IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, pages 194–200. IEEE, 2005. 3, 119

[19] Haiyan Yin, Linbo Luo, Wentong Cai, Yew-Soon Ong, and Jinghui Zhong. A data-driven approach for online adaptation of game difficulty. In *2015 IEEE conference on computational intelligence and games (CIG)*, pages 146–153. IEEE, 2015. 2

[20] MM Hassan Mahmud, Benjamin Rosman, Subramanian Ramamoorthy, and Pushmeet Kohli. Adapting interaction environments to diverse users through online action set selection. In *Workshops at the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014. 2, 125

[21] Avi Segal, Kobi Gal, Ece Kamar, Eric Horvitz, and Grant Miller. Optimizing interventions via offline policy evaluation: Studies in citizen science. 2018. 3, 90, 125, 126

[22] Shitian Shen and Min Chi. Reinforcement learning: the sooner the better, or the later the better? In *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization*, pages 37–44. ACM, 2016. 3

[23] Avi Segal, Yossi Ben David, Joseph Jay Williams, Kobi Gal, and Yaar Shalom. Combining difficulty ranking with multi-armed bandits to sequence educational content. In *International Conference on Artificial Intelligence in Education*, pages 317–321. Springer, 2018. 3, 87, 89, 90, 91, 104, 121

[24] Wim J Van der Linden, Cees AW Glas, et al. *Computerized adaptive testing: Theory and practice*. Springer, 2000. 3

[25] Julian Togelius, Renzo De Nardi, and Simon M Lucas. Towards automatic personalised content creation for racing games. In *2007 IEEE Symposium on Computational Intelligence and Games*, pages 252–259. IEEE, 2007. 3

[26] Martin Jennings-Teats, Gillian Smith, and Noah Wardrip-Fruin. Polymorph: dynamic difficulty adjustment through level generation. In *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*, page 11. ACM, 2010. 3

[27] Mihaly Csikszentmihalyi. Toward a psychology of optimal experience. In *Flow and the foundations of positive psychology*, pages 209–226. Springer, 2014. 3, 39

[28] Lev Vygotsky. Interaction between learning and development. *Readings on the development of children*, 23(3):34–41, 1978. 3, 16, 33

[29] Gustavo Danzi, Andrade Hugo Pimentel Santana, André Wilson Brotto Furtado, André Roberto Gouveia, Amaral Leitao, and Geber Lisboa Ramalho. Online adaptation of computer games agents: A reinforcement learning approach. In *II Workshop de Jogos e Entretenimento Digital*, pages 105–112, 2003. 3

[30] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 5

[31] George E Dahl, Dong Yu, Li Deng, and Alex Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on audio, speech, and language processing*, 20(1):30–42, 2011. 5

[32] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015. 5, 90, 93, 98, 122

[33] George Tucker, Surya Bhupatiraju, Shixiang Gu, Richard E Turner, Zoubin Ghahramani, and Sergey Levine. The mirage of action-dependent baselines in reinforcement learning. *arXiv preprint arXiv:1802.10031*, 2018. 5, 90, 93, 98, 122

[34] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354, 2017. 5

[35] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015. 90

[36] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937, 2016. 5, 93

[37] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897, 2015. 5, 90

[38] Jerry Alan Fails and Dan R Olsen Jr. Interactive machine learning. In *Proceedings of the 8th international conference on Intelligent user interfaces*, pages 39–45. ACM, 2003. 5

[39] Todd Kulesza, Margaret Burnett, Weng-Keen Wong, and Simone Stumpf. Principles of explanatory debugging to personalize interactive machine learning. In *Proceedings of the 20th international conference on intelligent user interfaces*, pages 126–137. ACM, 2015.

[40] Mayank Kabra, Alice A Robie, Marta Rivera-Alba, Steven Branson, and Kristin Branson. Jaaba: interactive machine learning for automatic annotation of animal behavior. *Nature methods*, 10(1):64, 2013. 5

[41] Francisco Cruz, Sven Magg, Cornelius Weber, and Stefan Wermter. Training agents with interactive reinforcement learning and contextual affordances. *IEEE Transactions on Cognitive and Developmental Systems*, 8(4):271–284, 2016. 5

[42] Andreas Holzinger. Interactive machine learning for health informatics: when do we need the human-in-the-loop? *Brain Informatics*, 3(2):119–131, 2016. 5

[43] Yaqian Zhang and Wooi Boon Goh. The influence of peer accountability on attention during gameplay. *Computers in Human Behavior*, 84:18–28, 2018. 9

[44] Colin M MacLeod. Half a century of research on the stroop effect: an integrative review. *Psychological bulletin*, 109(2):163, 1991. 9, 11

[45] Colin M MacLeod. The stroop task: The" gold standard" of attentional measures. *Journal of Experimental Psychology: General*, 121(1):12, 1992. 9, 14

[46] Wei Peng and Gary Hsieh. The influence of competition, cooperation, and player relationship in a motor performance centered computer game. *Computers in Human Behavior*, 28(6):2100–2106, 2012. 9, 14, 15, 22, 31, 32

[47] Kristin Siu, Alexander Zook, and Mark O Riedl. Collaboration versus competition: Design and evaluation of mechanics for games with a purpose. In *FDG*, 2014. 10, 14, 22

[48] Magy Seif El-Nasr, Bardia Aghabeigi, David Milam, Mona Erfani, Beth Lameman, Hamid Maygoli, and Sang Mah. Understanding and evaluating cooperative games. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 253–262. ACM, 2010. 10, 35, 36

[49] Ling Yuan, Yanhong Tu, Jian Li, and Lutao Ning. The impact of team ability disparity and reward structure on performance. *Systems Research and Behavioral Science*, 35(1):114–126, 2018. 10, 16, 34

[50] Kirstie Hawkey, Melanie Kellar, Derek Reilly, Tara Whalen, and Kori M Inkpen. The proximity factor: impact of distance on co-located collaboration. In *Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work*, pages 31–40. ACM, 2005. 10, 16

[51] Angela T Hall, Dwight D Frink, and M Ronald Buckley. An accountability account: A review and synthesis of the theoretical and empirical research on felt accountability. *Journal of Organizational Behavior*, 38(2):204–224, 2017. 10, 13

[52] Jennifer S Lerner and Philip E Tetlock. Accounting for the effects of accountability. *Psychological bulletin*, 125(2):255, 1999. 10, 13

[53] Philip E Tetlock. Accountability: The neglected social context of judgment and choice. *Research in organizational behavior*, 7(1):297–332, 1985. 10, 13

[54] Angela T Hall and Gerald R Ferris. Accountability and extra-role behavior. *Employee Responsibilities and Rights Journal*, 23(2):131–144, 2011. 10, 12

[55] Robert Folger and Russell Cropanzano. Fairness theory: Justice as accountability. *Advances in organizational justice*, 1:1–55, 2001. 12

[56] Dwight D Frink and Richard J Klimoski. Toward a theory of accountability in organizations and human resource management. 1998. 10, 12

[57] Paul A OKeefe, EJ Horberg, and Isabelle Plante. The multifaceted role of interest in motivation and engagement. In *The science of interest*, pages 49–67. Springer, 2017. 11

[58] Charlene Jennett, Anna L Cox, Paul Cairns, Samira Dhoparee, Andrew Epps, Tim Tijs, and Alison Walton. Measuring and defining the experience of immersion in games. *International journal of human-computer studies*, 66 (9):641–661, 2008. 11

[59] Suzanne De Castell and Jennifer Jenson. Paying attention to attention: New economies for learning. *Educational Theory*, 54(4):381–397, 2004. 11

[60] Philip E Tetlock, Linda Skitka, and Richard Boettger. Social and cognitive strategies for coping with accountability: Conformity, complexity, and bolstering. *Journal of personality and social psychology*, 57(4):632, 1989. 12, 13

[61] David W Johnson and Roger T Johnson. An educational psychology success story: Social interdependence theory and cooperative learning. *Educational researcher*, 38(5):365–379, 2009. 13

[62] Stephen E Lanivich, Jeremy R Brees, Wayne A Hochwarter, and Gerald R Ferris. Pe fit as moderator of the accountability–employee reactions relationships: Convergent results across two samples. *Journal of Vocational Behavior*, 77(3):425–436, 2010. 13

[63] Anne R Fitzpatrick and Alice H Eagly. Anticipatory belief polarization as a function of the expertise of a discussion partner. *Personality and Social Psychology Bulletin*, 7(4):636–642, 1981. 13

[64] Jan L Plass, Paul A O'Keefe, Bruce D Homer, Jennifer Case, Elizabeth O Hayward, Murphy Stein, and Ken Perlin. The impact of individual, competitive, and collaborative mathematics game play on learning, performance, and motivation. *Journal of educational psychology*, 105(4):1050, 2013. 14, 16, 22, 31

[65] Sarah Tausch, Stephanie Ta, and Heinrich Hussmann. A comparison of cooperative and competitive visualizations for co-located collaboration. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 5034–5039. ACM, 2016. 14, 17

[66] ClJ Bench, CD Frith, PM Grasby, KJ Friston, E Paulesu, RSJ Frackowiak, and Raymond J Dolan. Investigations of the functional anatomy of attention using the stroop test. *Neuropsychologia*, 31(9):907–922, 1993. 14

[67] Arwen M Marker and Amanda E Staiano. Better together: outcomes of cooperation versus competition in social exergaming. *Games for health journal*, 4(1):25–30, 2015. 15

[68] Amanda Roy and Christopher J Ferguson. Competitively versus cooperatively? an analysis of the effect of game play on levels of stress. *Computers in Human Behavior*, 56:14–20, 2016. 15

[69] Adam Lobel, Rutger CME Engels, Lisanne L Stone, and Isabela Granic. Gaining a competitive edge: Longitudinal associations between childrens competitive video game playing, conduct problems, peer relations, and prosocial behavior. *Psychology of Popular Media Culture*, 8(1):76, 2019. 15

[70] Guillaume Chanel, J Matias Kivikangas, and Niklas Ravaja. Physiological compliance for social gaming analysis: Cooperative versus competitive play. *Interacting with Computers*, 24(4):306–316, 2012. 15

[71] Mary Beth Stanne, David W Johnson, and Roger T Johnson. Does competition enhance or inhibit motor performance: A meta-analysis. *Psychological bulletin*, 125(1):133, 1999. 15

[72] Rory McGloin, Kyle S Hull, and John L Christensen. The social implications of casual online gaming: Examining the effects of competitive setting and performance outcome on player perceptions. *Computers in Human Behavior*, 59:173–181, 2016. 15

[73] Julia Crouse Waddell and Wei Peng. Does it matter with whom you slay? the effects of competition, cooperation and relationship type among video game players. *Computers in Human Behavior*, 38:331–338, 2014. 16

[74] Nicolas Nova. A review of how space affords socio-cognitive processes during collaboration. *PsychNology*, 3(ARTICLE):118–148, 2005. 16, 35

[75] Paul Light and Karen Littleton. Peer interaction and learning: perspectives and starting points. *Social Processes in childrens learning, Cambridge University Press, UK*, pages 1–13, 2000. 16

[76] Marlene Scardamalia and Carl Bereiter. Higher levels of agency for children in knowledge building: A challenge for the design of new knowledge media. *The Journal of the learning sciences*, 1(1):37–68, 1991. 16

[77] Susan Creighton and Andrea Szymkowiak. The effects of cooperative and competitive games on classroom interaction frequencies. *Procedia-Social and Behavioral Sciences*, 140:155–163, 2014. 16

[78] Robert E Slavin, Marshall B Leavey, and Nancy A Madden. Combining cooperative learning and individualized instruction: Effects on student mathematics achievement, attitudes, and behaviors. *The Elementary School Journal*, 84(4):409–422, 1984. 16

[79] Robert E Slavin. Cooperative learning and academic achievement: Why does groupwork work?.[aprendizaje cooperativo y rendimiento académico:¿ por qué funciona el trabajo en grupo?]. *Anales de psicología/annals of psychology*, 30(3):785–791, 2014. 16

[80] Yiping Lou, Philip C Abrami, John C Spence, Catherine Poulsen, Bette Chambers, and Sylvia dApollonia. Within-class grouping: A meta-analysis. *Review of educational research*, 66(4):423–458, 1996. 16

[81] Dejana Mullins, Nikol Rummel, and Hans Spada. Are two heads always better than one? differential effects of collaboration on students computer-supported learning in mathematics. *International Journal of Computer-Supported Collaborative Learning*, 6(3):421–443, 2011. 16

[82] Shih-Wen Huang and Wai-Tat Fu. Don't hide in the crowd!: increasing social transparency between peer workers improves crowdsourcing outcomes. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 621–630. ACM, 2013. 17

[83] Bianca Beersma, John R Hollenbeck, Stephen E Humphrey, Henry Moon, Donald E Conlon, and Daniel R Ilgen. Cooperation, competition, and team performance: Toward a contingency approach. *Academy of Management Journal*, 46(5):572–590, 2003. 17

[84] José P Zagal, Jochen Rick, and Idris Hsi. Collaborative games: Lessons learned from board games. *Simulation & Gaming*, 37(1):24–40, 2006. 17

[85] John M Tauer and Judith M Harackiewicz. The effects of cooperation and competition on intrinsic motivation and performance. *Journal of personality and social psychology*, 86(6):849, 2004. 18

[86] Sarah N Mattson, Amy M Goodman, Chip Caine, Dean C Delis, and Edward P Riley. Executive functioning in children with heavy prenatal alcohol exposure. *Alcoholism: Clinical and Experimental Research*, 23(11): 1808–1815, 1999. 21

[87] Mikkel R Jakobsen and Kasper Hornbæk. Up close and personal: Collaborative work on a high-resolution multitouch wall display. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 21(2):11, 2014. 27

[88] Florian Mueller, Sophie Stellmach, Saul Greenberg, Andreas Dippon, Susanne Boll, Jayden Garner, Rohit Khot, Amani Naseem, and David Altimira. Proxemics play: understanding proxemics for designing digital play experiences. In *Proceedings of the 2014 conference on Designing interactive systems*, pages 533–542. ACM, 2014. 27

[89] Garriy Shteynberg. Shared attention at the origin: On the psychological power of descriptive norms. *Journal of Cross-Cultural Psychology*, 46(10): 1245–1251, 2015. 31

[90] Garriy Shteynberg, Jacob B Hirsh, Evan P Apfelbaum, Jeff T Larsen, Adam D Galinsky, and Neal J Roese. Feeling more together: Group attention intensifies emotion. *Emotion*, 14(6):1102, 2014. 31

[91] Herman A Witkin and Donald R Goodenough. Cognitive styles: essence and origins. field dependence and field independence. *Psychological issues*, (51): 1–141, 1981. 33

[92] Otto Köhler. Über den gruppenwirkungsgrad der menschlichen körperarbeit und die bedingung optimaler kollektivkraftreaktion. *Industrielle Psychotechnik*, 1927. 33, 34, 119

[93] Norbert L Kerr and Guido Hertel. The köhler group motivation gain: How to motivate the weak links in a group. *Social and Personality Psychology Compass*, 5(1):43–55, 2011. 33

[94] Guido Hertel, Norbert L Kerr, and Lawrence A Messé. Motivation gains in performance groups: Paradigmatic and theoretical developments on the köhler effect. *Journal of personality and social psychology*, 79(4):580, 2000. 34, 119

[95] Brandon C Irwin, Jennifer Scorniaenchi, Norbert L Kerr, Joey C Eisenmann, and Deborah L Feltz. Aerobic exercise is promoted when individual performance affects the group: a test of the kohler motivation gain effect. *Annals of Behavioral Medicine*, 44(2):151–159, 2012.

[96] Lawrence A Messé, Guido Hertel, Norbert L Kerr, Robert B Lount Jr, and Ernest S Park. Knowledge of partner's ability as a moderator of group motivation gains: An exploration of the köhler discrepancy effect. *Journal of Personality and social Psychology*, 82(6):935, 2002. 34, 119

[97] Sara Kiesler and Jonathon N Cummings. What do we know about proximity and distance in work groups? a legacy of research. *Distributed work*, 1:57–80, 2002. 35

[98] HH Clark, SE Brennan, LB Resnick, JM Levine, and SD Teasley. Grounding in communication perspectives on socially shared cognition (pp. 127–149). *Washington, DC, US: American Psychological Association*, 1991. 35

[99] Paul Dourish and Victoria Bellotti. Awareness and coordination in shared workspaces. In *CSCW*, volume 92, pages 107–114, 1992. 35

[100] José Bernardo Rocha, Samuel Mascarenhas, and Rui Prada. Game mechanics for cooperative games. *ZON Digital Games 2008*, pages 72–80, 2008. 35, 36

[101] Guillaume Chanel, Cyril Rebetez, Mireille Bétrancourt, and Thierry Pun. Boredom, engagement and anxiety as indicators for adaptation to difficulty in games. In *Proceedings of the 12th international conference on Entertainment and media in the ubiquitous era*, pages 13–17. ACM, 2008. 39, 40

[102] Oto Vozár and Mária Bieliková. Adaptive test question selection for web-based educational system. In *2008 Third International Workshop on Semantic Media Adaptation and Personalization*, pages 164–169. IEEE, 2008. 40

[103] Jan Papoušek and Radek Pelánek. Impact of adaptive educational system behaviour on student motivation. In *International Conference on Artificial Intelligence in Education*, pages 348–357. Springer, 2015. 39, 40, 90, 122, 125

[104] Pernille J Olesen, Helena Westerberg, and Torkel Klingberg. Increased prefrontal and parietal activity after training of working memory. *Nature neuroscience*, 7(1):75, 2004. 40, 41

[105] George A Alvarez and Patrick Cavanagh. The capacity of visual short-term memory is set both by visual information load and by number of objects. *Psychological science*, 15(2):106–111, 2004. 40, 42

[106] H Lee Swanson. Working memory, attention, and mathematical problem solving: A longitudinal study of elementary school children. *Journal of Educational Psychology*, 103(4):821, 2011. 41

[107] Torkel Klingberg, Elisabeth Fernell, Pernille J Olesen, Mats Johnson, Per Gustafsson, Kerstin Dahlström, Christopher G Gillberg, Hans Forssberg, and Helena Westerberg. Computerized training of working memory in children with adhd-a randomized, controlled trial. *Journal of the American Academy of Child & Adolescent Psychiatry*, 44(2):177–186, 2005. 41

[108] Mark D Rapport, Sarah A Orban, Michael J Kofler, and Lauren M Friedman. Do programs designed to train working memory, other executive functions, and attention benefit children with adhd? a meta-analytic review of cognitive, academic, and behavioral outcomes. *Clinical psychology review*, 33(8): 1237–1252, 2013. 41

[109] Joni Holmes, Susan E Gathercole, and Darren L Dunning. Adaptive training leads to sustained enhancement of poor working memory in children. *Developmental science*, 12(4):F9–F15, 2009. 41

[110] Edward K Vogel and Maro G Machizawa. Neural activity predicts individual differences in visual working memory capacity. *Nature*, 428(6984):748, 2004. 42

[111] Steven J Luck and Edward K Vogel. The capacity of visual working memory for features and conjunctions. *Nature*, 390(6657):279, 1997.

[112] Paul M Bays and Masud Husain. Dynamic shifts of limited working memory resources in human vision. *Science*, 321(5890):851–854, 2008. 42

[113] Yaoda Xu and Marvin M Chun. Visual grouping in human parietal cortex. *Proceedings of the national academy of sciences*, 104(47):18766–18771, 2007. 42

[114] Timothy F Brady, Talia Konkle, and George A Alvarez. A review of visual memory capacity: Beyond individual items and toward structured representations. *Journal of vision*, 11(5):4–4, 2011. 42

[115] Guy Shani and Bracha Shapira. Edurank: A collaborative filtering approach to personalization in e-learning. *Educational data mining*, pages 68–75, 2014. 44, 54, 73, 90, 120, 122

[116] Yaqian Zhang, Jacek Mańdziuk, Chai Hiok Quek, and Boon Wooi Goh. Curvature-based method for determining the number of clusters. *Information Sciences*, 415:414–428, 2017. 53

[117] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967. 54

[118] John Shawe-Taylor and Nello Cristianini. *Kernel methods for pattern analysis*. Cambridge university press, 2004.

[119] Enmei Tu, Longbing Cao, Jie Yang, and Nicola Kasabov. A novel graph-based k-means for nonlinear manifold clustering and representative selection. *Neurocomputing*, 143:109–122, 2014. 54

[120] Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974. 54, 57, 63, 133

[121] John A Hartigan and JA Hartigan. *Clustering algorithms*, volume 209. Wiley New York, 1975. 57, 63, 133

[122] Wojtek J Krzanowski and YT Lai. A criterion for determining the number of groups in a data set using sum-of-squares clustering. *Biometrics*, pages 23–34, 1988. 54, 57, 63, 133

[123] Stan Salvador and Philip Chan. Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In *Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on*, pages 576–584. IEEE, 2004. 55, 56, 121

[124] Catherine A Sugar and Gareth M James. Finding the number of clusters in a dataset: An information-theoretic approach. *Journal of the American Statistical Association*, 98(463):750–763, 2003. 55, 56, 58, 64, 65, 134

[125] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001. 55, 57, 63, 122, 134

[126] Cyril Goutte, Peter Toft, Egill Rostrup, Finn Å Nielsen, and Lars Kai Hansen. On clustering fmri time series. *NeuroImage*, 9(3):298–310, 1999. 56, 121

[127] Glenn W Milligan and Martha C Cooper. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50(2): 159–179, 1985. 57

[128] FHC Marriott. Practical problems in a method of cluster analysis. *Biometrics*, 27(3):501–514, 1971. 57

[129] Mark Ming-Tso Chiang and Boris Mirkin. Intelligent choice of the number of clusters in k-means clustering: an experimental study with different cluster spreads. *Journal of classification*, 27(1):3–40, 2010. 57, 65

[130] Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons, 2009. 57, 63, 122, 134

[131] Katherine S Pollard and Mark J Van Der Laan. A method to identify significant clusters in gene expression data. *World Multiconference on Systemics, Cybernetics and Informatics*, 5(2):318–325, 2002. 57

[132] Wei Fu and Patrick O Perry. Estimating the number of clusters using cross-validation. *arXiv preprint arXiv:1702.02658*, 2017. 57

[133] Alex Rodriguez and Alessandro Laio. Clustering by fast search and find of density peaks. *Science*, 344(6191):1492–1496, 2014. 58

[134] Kadim Tasdemir, Pavel Milenov, and Brooke Tapsall. Topology-based hierarchical clustering of self-organizing maps. *IEEE Transactions on Neural Networks*, 22(3):474–485, 2011. 58

[135] M. Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`. 59, 136

[136] Enmei Tu, Jie Yang, Nicola Kasabov, and Yaqian Zhang. Posterior distribution learning (pdl): A novel supervised learning framework using unlabeled samples to improve classification performance. *Neurocomputing*, 157:173–186, 2015. 59

[137] Enmei Tu, Yaqian Zhang, Lin Zhu, Jie Yang, and Nikola Kasabov. A graph-based semi-supervised k nearest-neighbor method for nonlinear manifold distributed data classification. *Information Sciences*, 367:673–688, 2016. 59

[138] Byron P Roe, Hai-Jun Yang, Ji Zhu, Yong Liu, Ion Stancu, and Gordon McGregor. Boosted decision trees as an alternative to artificial neural networks for particle identification. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 543(2):577–584, 2005. 70, 136

[139] Rajen B Bhatt, Gaurav Sharma, Abhinav Dhall, and Santanu Chaudhury. Efficient skin region segmentation using low complexity fuzzy decision tree model. In *2009 Annual IEEE India Conference*, pages 1–4. IEEE, 2009. 70, 136

[140] YY Yao. Measuring retrieval effectiveness based on user preference of documents. *Journal of the American Society for Information Science*, 46(2): 133–145, 1995. 73, 120

[141] Guy Shani and Asela Gunawardana. Evaluating recommendation systems. In *Recommender systems handbook*, pages 257–297. Springer, 2011. 73, 120

[142] John Stamper and Zachary A Pardos. The 2010 kdd cup competition dataset: Engaging the machine learning community in predictive learning analytics. *Journal of Learning Analytics*, 3(2):312–316, 2016. 84

[143] Yaqian Zhang and Wooi-Boon Goh. Bootstrapped policy gradient for difficulty adaptation in intelligent tutoring systems. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 711–719. International Foundation for Autonomous Agents and Multiagent Systems, 2019. 87

[144] Min Chi, Kurt VanLehn, Diane Litman, and Pamela Jordan. An evaluation of pedagogical tutorial tactics for a natural language tutoring system: A reinforcement learning approach. *International Journal of Artificial Intelligence in Education*, 21(1-2):83–113, 2011. 90

[145] Yun-En Liu, Travis Mandel, Emma Brunskill, and Zoran Popovic. Trading off scientific knowledge and user learning with multi-armed bandits. In *EDM*, pages 161–168, 2014. 90

[146] Benjamin Clement, Didier Roy, Pierre-Yves Oudeyer, and Manuel Lopes. Multi-armed bandits for intelligent tutoring systems. *Journal of Educational Data Mining*, 7(2), 2015. 90, 91, 104

[147] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Van Hasselt, Marc Lanctot, and Nando De Freitas. Dueling network architectures for deep reinforcement learning. *arXiv preprint arXiv:1511.06581*, 2015. 90

[148] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015. 90

[149] Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. In *Advances in neural information processing systems*, pages 2249–2257, 2011. 90

[150] Ganesh Ghalme, Shweta Jain, Sujit Gujar, and Y Narahari. Thompson sampling based mechanisms for stochastic multi-armed bandit problems. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pages 87–95. International Foundation for Autonomous Agents and Multiagent Systems, 2017.

[151] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. In *Advances in neural information processing systems*, pages 4026–4034, 2016. 90

[152] Kamil Ciosek and Shimon Whiteson. Expected policy gradients. *arXiv preprint arXiv:1706.05374*, 2017. 90

[153] Richard S Sutton and Andrew G Barto. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 1998. 90, 93

[154] Shixiang Gu, Timothy Lillicrap, Zoubin Ghahramani, Richard E Turner, and Sergey Levine. Q-prop: Sample-efficient policy gradient with an off-policy critic. *arXiv preprint arXiv:1611.02247*, 2016. 90, 93, 98, 122

[155] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992. 90, 93, 98, 107, 122

[156] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000. 90, 92, 102

[157] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 5026–5033. IEEE, 2012. 90, 107

[158] Cathy Wu, Aravind Rajeswaran, Yan Duan, Vikash Kumar, Alexandre M Bayen, Sham Kakade, Igor Mordatch, and Pieter Abbeel. Variance reduction for policy gradient with action-dependent factorized baselines. *arXiv preprint arXiv:1803.07246*, 2018. 93, 98

[159] Hao Liu, Yihao Feng, Yi Mao, Dengyong Zhou, Jian Peng, and Qiang Liu. Action-dependent control variates for policy optimization via stein identity. 2018. 93, 98, 122

[160] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *ICML*, 2014. 102, 103, 107

[161] Ronald K Hambleton, Hariharan Swaminathan, and H Jane Rogers. *Fundamentals of item response theory*, volume 2. Sage, 1991. 104

[162] Alexander Shkolnik and Russ Tedrake. Path planning in 1000+ dimensions using a task-space voronoi bias. In *2009 IEEE International Conference on Robotics and Automation*, pages 2061–2067. IEEE, 2009. 107

[163] Travis Mandel, Yun-En Liu, Emma Brunskill, and Zoran Popovic. Offline evaluation of online reinforcement learning algorithms. In *AAAI*, pages 1926–1933, 2016. 125, 126

[164] Lukasz A Kurgan, Krzysztof J Cios, Ryszard Tadeusiewicz, Marek Ogiela, and Lucy S Goodenday. Knowledge discovery approach to automated cardiac spect diagnosis. *Artificial intelligence in medicine*, 23(2):149–169, 2001. 136

[165] W Nick Street, William H Wolberg, and Olvi L Mangasarian. Nuclear feature extraction for breast tumor diagnosis. In *IS&T/SPIE's Symposium on Electronic Imaging: Science and Technology*, pages 861–870. International Society for Optics and Photonics, 1993. 136

[166] Max A Little, Patrick E McSharry, Stephen J Roberts, Declan AE Costello, and Irene M Moroz. Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection. *BioMedical Engineering OnLine*, 6(1):1, 2007. 136

[167] Shelby J Haberman. Generalized residuals for log-linear models. In *Proceedings of the 9th international biometrics conference*, pages 104–122, 1976. 136

[168] I-Cheng Yeh, King-Jang Yang, and Tao-Ming Ting. Knowledge discovery on rfm model using bernoulli sequence. *Expert Systems with Applications*, 36 (3):5866–5871, 2009. 136

[169] RK Bock, A Chilingarian, M Gaug, F Hakl, Th Hengstebeck, M Jiřina, J Klaschka, E Kotrč, P Savickỳ, S Towers, et al. Methods for multidimensional event classification: a case study using images from a cherenkov gamma-ray telescope. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 516(2):511–528, 2004. 136

[170] Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89(1):31–71, 1997. 136

[171] Franz Oppacher Lee Graham. Hill-valley data set, 2016. URL `https://archive.ics.uci.edu/ml/datasets/Hill-Valley`. 136

[172] Diogo Ayres-de Campos, Joao Bernardes, Antonio Garrido, Joaquim Marques-de Sa, and Luis Pereira-Leite. Sisporto 2.0: a program for automated analysis of cardiotocograms. *Journal of Maternal-Fetal Medicine*, 9 (5):311–318, 2000. 136

[173] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936. 136

[174] Paulo Cortez, António Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4):547–553, 2009. 136