A DATA-DRIVEN APPROACH FOR PERSONALIZED DRAMA MANAGEMENT

A Thesis
Presented to
The Academic Faculty

by

Hong Yu

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Computer Science

Georgia Institute of Technology
August 2015

A DATA-DRIVEN APPROACH FOR PERSONALIZED DRAMA MANAGEMENT

Approved by:

Dr. Mark O. Riedl, Advisor
School of Interactive Computing
Georgia Institute of Technology

Dr. Charles Isbell
School of Interactive Computing
Georgia Institute of Technology

Dr. Brian Magerko
School of Literature, Media, and Communication
Georgia Institute of Technology

Dr. David Roberts
Department of Computer Science
North Carolina State University

Dr. Andrea Thomaz
School of Interactive Computing
Georgia Institute of Technology

Date Approved: April 23, 2015

To my family

ACKNOWLEDGEMENTS

First and foremost, I would like to express my most sincere gratitude and appreciation to Mark Riedl, who has been my advisor and mentor throughout the development of my study and research at Georgia Tech. He has been supportive ever since the days I took his Advanced Game AI class and entered the Entertainment Intelligence lab. Thanks to him I had the opportunity to work on the interactive narrative project which turned into my thesis topic. Without his consistent guidance, encouragement and support, this dissertation would never have been successfully completed.

I would also like to gratefully thank my dissertation committee, Charles Isbell, Brian Magerko, David Roberts and Andrea Thomaz for their time, effort and the opportunities to work with them. Their expertise, insightful comments and experience in multiple research fields have been really beneficial to my thesis research.

A lot of friends and colleagues have been generous and helpful to make my study fun and memorable in the entertainment intelligence lab. Many thanks go to Boyang Li, Alexander Zook, Brian O'Neill, Kristin Siu, Matthew Guzdial, Brent Harrison, Stephen Lee-Urban, Yangfeng Ji, and Spencer Frazier.

I would also like to wish my thanks to Jessica Celestine, Cynthia Bryant, and everyone in the School of Interactive Computing for their help and support during my study at Georgia Tech.

Finally, I would like to thank my family for always being supportive and standing by me through good times and bad. I deeply thank for their love, support, and reassurances in all my pursuits.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

An interactive narrative is a form of digital entertainment in which players can create or influence a dramatic storyline through actions, typically by assuming the role of a character in a fictional virtual world. The interactive narrative systems usually employ a drama manager (DM), an omniscient background agent that monitors the fictional world and determines what will happen next in the players' story experience. Prevailing approaches to drama management choose successive story plot points based on a set of criteria given by the game designers. In other words, the DM is a surrogate for the game designers.

In this dissertation, I create a data-driven personalized drama manager that takes into consideration players' preferences. The personalized drama manager is capable of (1) modeling the players' preference over successive plot points from the players' feedback; (2) guiding the players towards selected plot points without sacrificing players' agency; (3) choosing target successive plot points that simultaneously increase the player's story preference ratings and the probability of the players selecting the plot points.

To address the first problem, I develop a collaborative filtering algorithm that takes into account the specific sequence (or history) of experienced plot points when modeling players' preferences for future plot points. Unlike the traditional collaborative filtering algorithms that make one-shot recommendations of complete story artifacts (e.g., books, movies), the collaborative filtering algorithm I develop is a sequential recommendation algorithm that makes every successive recommendation based on all previous recommendations. To address the second problem, I create a

multi-option branching story graph that allows multiple options to point to each plot point. The personalized DM working in the multi-option branching story graph can influence the players to make choices that coincide with the trajectories selected by the DM, while gives the players the full agency to make any selection that leads to any plot point in their own judgement. To address the third problem, the personalized DM models the probability that the players transitioning to each full-length stories and selects target stories that achieve the highest expected preference ratings at every branching point in the story space.

The personalized DM is implemented in an interactive narrative system built with choose-your-own-adventure stories. Human study results show that the personalized DM can achieve significantly higher preference ratings than non-personalized DMs or DMs with pre-defined player types, while preserve the players' sense of agency.

CHAPTER I

INTRODUCTION

Narratives, in the form of oral, written, visual or digital format, play a central role in many forms of entertainment media, including novels, movies, and theaters. Previous cognitive and psychological research suggests that narratives serve important cognitive and social functions, such as communication [8, 42], social interaction [16, 27], and learning [21], etc. In story-based computer games and other training and education systems, narratives work as a means to motivate player's activity and to create a sense of causal continuity across a series of challenges [57].

A narrative is a recounting of one or more real or fictitious events communicated by one or more narrators to one or more narratees [46]. For simplicity, the narrative can be broken up into a sequence of *plot points*. In this dissertation, a *plot point* is used to refer to a fictitious event that encapsulates one or more character actions, dialogue, or behaviors over a short period of time in the narrative. There are two fundamental types of narratives: linear narrative and branching narrative [50]. Linear narrative is the traditional form of narrative in which a sequence of plot points is narrated from beginning to ending without variation or possibility of a user altering the way in which the story unfolds or ends. Many computer games employ linear narratives. The story structure of the games is usually partitioned into levels. The outcome of each level is either successful completion or failure, in which case the player must start the level over. All players experience the same story throughout the game.

A branching narrative enables users to influence the way in which a narrative unfolds or ends, thus increasing user engagement. The users can make a selection or perform an action at certain predefined points in the narratives. The narrative will then unfold into one of a predefined set of alternative storyline continuations based on the users' selection. The structure of the branching narrative can be represented by a *branching story graph*, a directed acyclic graph in which each node represents a plot point, and arcs denote alternative choices of action that the player can choose [49, 74, 76, 75]. Any route from the root to any leaf node in the graph is a full-length story experience for the users. Branching story graphs are found in the choose-your-own-adventure (CYOA) series of novels[1], and also used to great effect in hypermedia and interactive systems [49].

The choose-your-own-adventure is a simple form of branching narrative that does not require artificial intelligence. The CYOA books contain adventure stories originally created for teenagers in the US. At the end of every page in the CYOA books, there are several options a reader can choose from. The reader will continue to read different pages based on the options he/she chooses and the story will unfold into a different branch. Figure 1 shows the branching story graph representation of one of the CYOA books—*The Abominable Snowman*. Each node in the graph represents a page in the book (a plot point). Every story starts from the root node and ends on one of the leaves. As shown in Figure 1, the branching story graph encodes human authorial intent since it specifies which plot points are allowed to follow other plot points.

An *interactive narrative* is a computer game style of branching narrative. The interactive narrative is a form of digital interactive experience in which players create or influence a dramatic storyline through actions, either by assuming the role of a character in a fictional virtual world, issuing commands to computer-controlled characters, or directly manipulating the fictional world state [51]. The structure of most interactive narrative can be represented by a branching story graph. The players are allowed to influence the storyline by performing actions or making selections at

[1] http://en.wikipedia.org/wiki/Choose˙Your˙Own˙Adventure

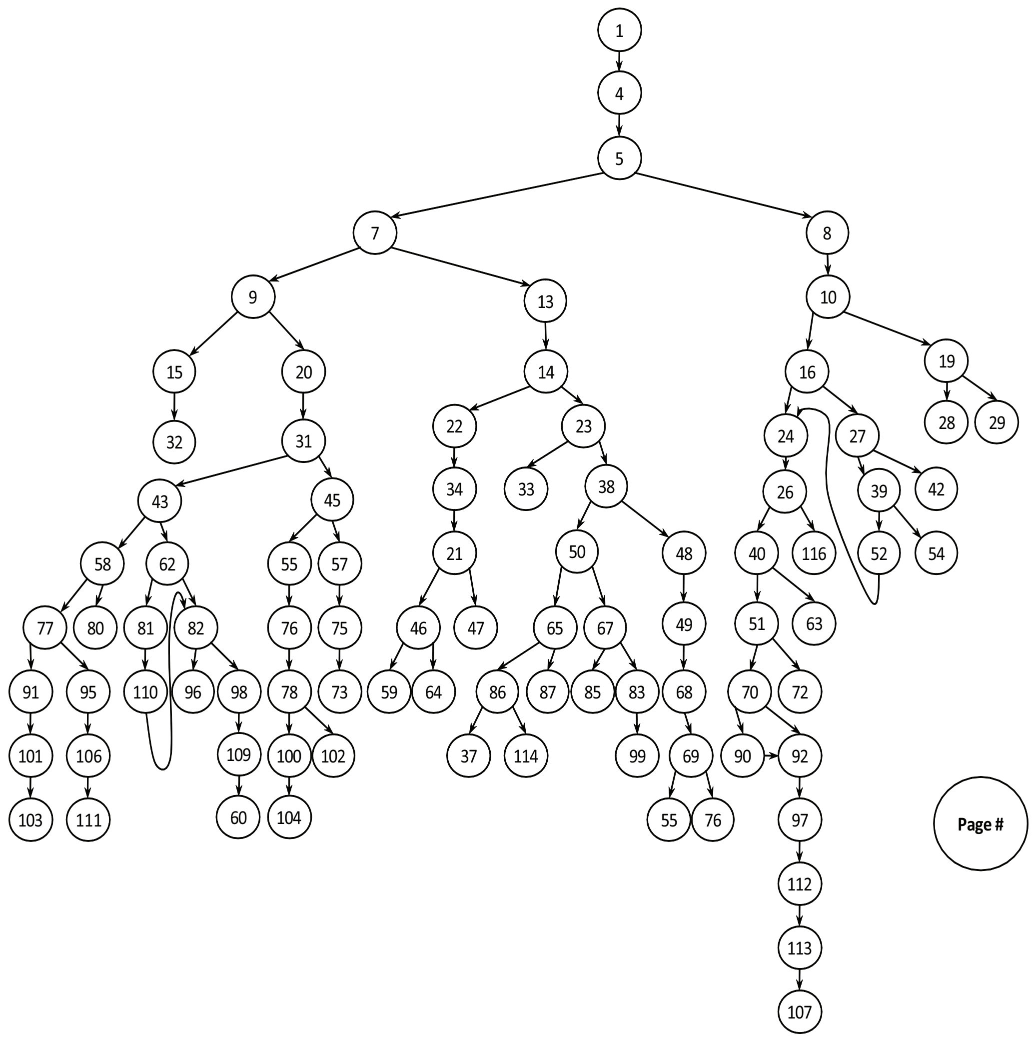

Figure 1: Illustration of the branching story graph for one of the CYOA books—*The Abominable Snowman.*

predefined branching points in the branching story graph. In this research, I focus on the interactive narrative and other computer games/tutoring systems of which the story structure can be represented by a branching story graph in which nodes are plot points[2].

Unlike the CYOA series, interactive narrative often employs a *Drama Manager* (DM) [4, 70], an omniscient background agent that monitors the fictional world and determines what will happen next in the player's story experience. The goal of the drama manager is to coordinate and/or instruct virtual characters in response to player actions, and achieve a coherent and enjoyable narrative experience while preserving the *player agency* in the virtual world. The player agency is the satisfying power to take meaningful action and see the results of the decisions and choices [37]. Preserving player agency means that the drama manager should at least allow the players to make choices by themselves in the interactive narrative. To enhance the players' experience through the branching story graph, the DM should also be able to evaluate possible future narrative sequences that the player could experience. At each plot point in the branching story graph, the DM must determine which successive plot points are the most appropriate for the current player and guide the player to the selected plot points without sacrificing player agency.

Prevailing approaches to drama management evaluate successive story plot points based on a set of criteria given by the game designer [39, 70, 55, 52, 31, 33]. This set of criteria is provided by the human author beforehand and is the only measures for the quality of the player's interactive experience. Thus the majority of prior approaches to drama management use the DM as a surrogate for human designer [74, 77].

I believe that different types of players have different opinions on whether a story

[2] The algorithms developed in this dissertation work on the level of the branching story graph. Although most role playing games can be represented by the branching story graph at some abstract level, the branching story graph may not capture all the detailed lower level player behaviors/interaction such as the players' social interaction with other characters, player emotion, etc. in the games.

experience is enjoyable or not. In this dissertation, I will build a drama manager that can model players and then use the player models to increase the quality of individual players' story experiences. That is, I aim to create a personalized drama manager that is *also* a surrogate for the player by taking into account the player's preferences.

Player modeling has been applied in drama management to adapt computer games [29]. But relatively little work has been done to build the drama manager that uses the player models to personalize story experience for players. Most previous approaches attempt to optimize player experience by classifying players according to well-defined player types [41, 12, 67, 62], and using pre-defined mappings of classes to plot point selection rules. These approaches require a designer to pre-determine the meaningful player types, even though there is no clear evidence of links between player type models and story preferences.

In this dissertation, I aim to improve players' subjective experience through building a data-driven personalized drama manager without pre-defined player types. The personalized DM is capable of discovering player types directly from player feedback and can be easily generalized to a different domain where pre-defined player types might not fit. It guides the players in a CYOA style interactive narrative system and maximizes the players' *expected* preference ratings for their story experience. In the remaining part of this chapter, I will describe my research questions in building the personalized DM, followed by my proposed solutions and my contributions.

1.1 Research Questions and Intended Contributions

To build the personalized drama manager that can increase the players' preference ratings for their story experience in interactive narratives, I propose the following three research questions (RQ) and thesis statement:

> **RQ1.** How can the drama manager model players' preference over successive plot points?

A story is a *sequence* of plot points. A player's assessment of the story one is experiencing is thus a function of the history of plot points experienced so far [56], instead of the current plot point. Many traditional machine learning algorithms for modeling user preferences do not take into account historical information, thus can not be adopted directly to model players' preference over successive plot point.

> **RQ2.** How can the drama manager guide players towards a selected plot point without sacrificing players' agency?

Player agency is a critical aspect of interactive narrative. Some drama management systems enforce certain narrative trajectories by limiting player agency in various ways [48, 73, 39, 52]. I aim to build a personalized DM that can increase the probability that the players choose the selected plot points but also give the players the rights to choose other successive plot points in CYOA style interactive narrative system.

> **RQ3.** How can the drama manager select story experience in interactive narrative system?

Even though the personalized DM builds an accurate player preference model and increases the probability the player transitioning to the plot points that are expected to achieve the highest predicted preference ratings, the players may still choose the options leading to other narrative experience because they are given the full agency to make their own choices. It is possible that the personalized DM finds no plot point that can lead to a player preferred narrative experience after it fails to guide a player at some branching point in the branching story graph.

Based on the above research questions, I pose the following thesis statement:

> **Thesis Statement.** A personalized drama manager, utilizing the sequential preference models dynamically built from the players' feedback and the personalized guidance algorithm, achieves significantly higher

self-reported preference ratings for their story experience than a non-personalized drama manager or a drama manager with pre-defined player types for interactive narrative systems.

In response to RQ1, I develop a technique to learn a data-driven player preference model using collaborative filtering (CF) instead of using pre-defined player types. Collaborative filtering has been successfully applied in recommender systems to model user preference over movies, books, music, and other products from users' structural feedback, e.g. ratings, previous purchase, etc [64]. The CF algorithms attempt to learn users' preference patterns from users' preference ratings and predict new user's ratings from previous users' ratings which share similar preference patterns. The player models built with collaborative filtering algorithms do not take any assumption about pre-defined player types. I propose to utilize CF algorithms to model players' preferences over the future plot point sequences based on the players' feedback. Unlike the traditional usage of CF for *one-shot* recommendations of complete story artifacts (e.g., books, movies), modeling preference over successive plot points is a new type of recommendation problem—*sequential recommendation*, in which each subsequent recommendation is dependent on the entire sequence of prior recommendations for a particular story experience. To address the sequential recommendation problem, I propose a prefix-based collaborative filtering (PBCF) algorithm. Chapter 3 will describe in details how the PBCF algorithm is used to build flexible player models from players' explicit feedback, i.e. ratings. The PBCF algorithm learns player preferences over fragments of story and then applies it to select successive plot points in a simple story generation system I built based on choose-your-own-adventure books. The story generation system is similar to an interactive narrative system except that it ignores player agency, which will be revisited as part of RQ2.

In response to RQ2, I propose a personalized guidance algorithm that manipulates the branching story graph such that the player is more likely to make a choice that coincides with the trajectory selected by the drama manager. The drama manager using the personalized guidance algorithm gives the players the full agency to select any option that leads to any plot point in their own judgement. Collaborative filtering is used to build player models to predict the players' preference over the options in the story space. Chapter 4 will describe in more details the proposed personalized guidance algorithm in interactive narrative systems.

In response to RQ3, I propose a personalized drama manager that can maximize the players' *expected* preference ratings in interactive narrative. The personalized drama manager models the player's preference over successive plot point sequences using PBCF algorithm, builds a player model to predict the probability that the player selects each option at each branching point, chooses a successive branch that simultaneously increases the player's story preference ratings and the probability of the player selecting the branch, and increases the probability that the player transitions to the selected branch using the personalized guidance algorithm. The personalized drama manager will be described in details in Chapter 4.

The major contributions of this thesis are:

- A data-driven player modeling algorithm that predicts players' preference over future plot points in interactive narrative.
- A personalized guidance algorithm that increases the likelihood that the players select the desired plot points in interactive narrative.
- A personalized drama manager that maximizes the players' self-reported story preference ratings in interactive narrative.

1.2 Reader's Guide

The remainder of this dissertation is arranged as follows: Chapter 2 describes related work on interactive narrative, player modeling and drama manager. Chapter 3 describes how the PBCF algorithm is used to build the player model from the players' explicit feedback—ratings, and predict players preference over future plot points in storytelling systems. Chapter 4 describes the personalized guidance algorithm to guide players towards the desired successive plot points and the personalized drama manager algorithm to increase the players' expected enjoyment ratings in interactive narrative systems. Chapter 5 describes two further improvements to the personalized drama manager: reducing the required amount of training data and incorporating author's preference into the personalized drama manager. Chapter 6 concludes the entire dissertation.

CHAPTER II

BACKGROUND AND RELATED WORK

The research problem I am addressing is how to build a personalized drama manager that increases players' story preference ratings in interactive narrative. Therefore, this chapter will first introduce the basic concept of narrative and interactive narrative, followed by some background information about the narrative structure I am going to use in this research—the branching story graph. Then I will shift to a discussion of previous work on various types of drama managers, which is what I am building. Player modeling is an essential part of my personalized drama manager. Thus I will also describe previous player modeling research in computer games, especially in interactive narrative. At the end, I will briefly review collaborative filtering algorithms and their application in computer games.

2.1 Narrative and Interactive Narrative

Narrative plays an important role in human culture. We use narrative to communicate, entertain, and teach, etc. Prince [46] defines narrative as:

Definition 1 (Narrative)**.** *A narrative is a recounting of one or more real or fictitious events communicated by one or more narrators to one or more narratees.*

Prince's definition indicates that a narrative can be presented as a sequence of one or more events, which are called plot points in this dissertation. Instead of a laundry list of random plot points, narrative has a continuant subject and constitutes a whole. The main plot points of a narrative form a plot, which usually adheres to a particular pattern, such as Freytag's triangle [14] and Aristotelian dramatic arcs [3]. Figure 2 shows a typical Aristotelian dramatic arc.

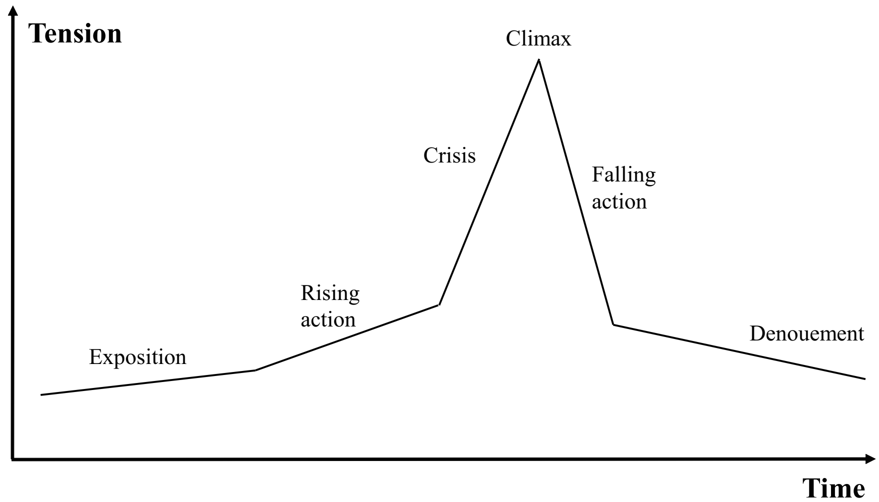

Figure 2: Aristotle's dramatic arc.

Traditional linear narratives are usually narrated from the begin to the end without the possibility for the narratees to change the progression or ending of the narratives. Branching narratives enable users to influence the way in which the narrative unfolds or ends. The users can make a selection or perform an action at predefined points to push the narrative into one of a predefined set of alternative storyline branches. An interactive narrative is a digital type of branching narrative in which the players can influence the storyline through interacting with the fictional world. Riedl and Bulitko [51] defines the interactive narrative as:

Definition 2 (Interactive Narrative)**.** *Interactive narrative is a form of digital interactive experience in which users create or influence a dramatic storyline through actions, either by assuming the role of a character in a fictional virtual world, issuing commands to computer-controlled characters, or directly manipulating the fictional world state.*

The structure of an interactive narrative can be represented by a branching story graph—a directed acyclic graph in which nodes represent story plot points and arcs denote alternative choices of action that the player can choose. The next section will introduce background information on the branching story graph.

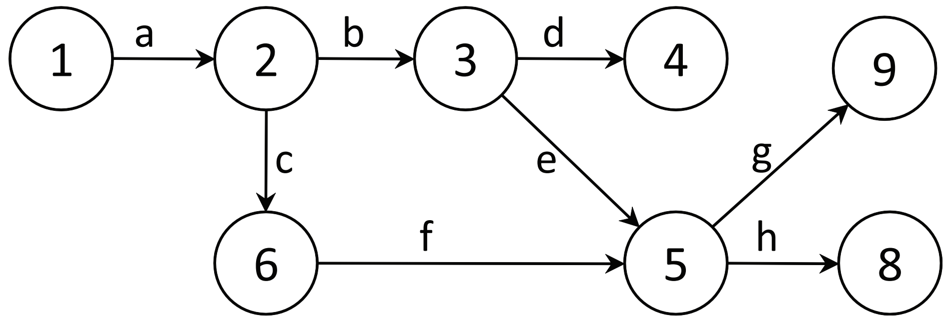

Figure 3: A sample branching story graph.

2.2 Branching Story Graph

Figure 3 shows a simple branching story graph; each round node is a plot point and links denote options that players can make during each plot point. Depending on which options a player choose, the Drama Manager will present the corresponding next plot point to the player. For example, in Figure 3, if the player is at node 3 and chooses option E, she will navigate to plot point 5 in the next step. A path through the graph starting from the root node and terminating at any leaf node is a possible complete story experience for the player. Figure 3 contains five possible complete stories: {1,2,3,4}, {1,2,3,5,8}, {1,2,3,5,9}, {1,2,6,5,8}, and {1,2,6,5,9}.

In this dissertation, I assume there exists a pre-authored branching story graph (including all the options), which is stored in the *story library*. The branching story graph may have been authored by hand or by some other intelligent process (c.f., [70, 49, 55, 39]) or through collaborative editing techniques such as crowdsourcing [28]. This assumption allows me to focus on the DM decision-making process and the development and validation of the player modeling techniques.

Although the concept of a branching story graph is simple, many other story representations used by AI-driven interactive narrative systems and story-based RPG games (such as planning based DMs and optimization/search based DMs) are reducible to the branching story graph [49, 74].

Riedl and Young have proved the correlation between the planning based narrative

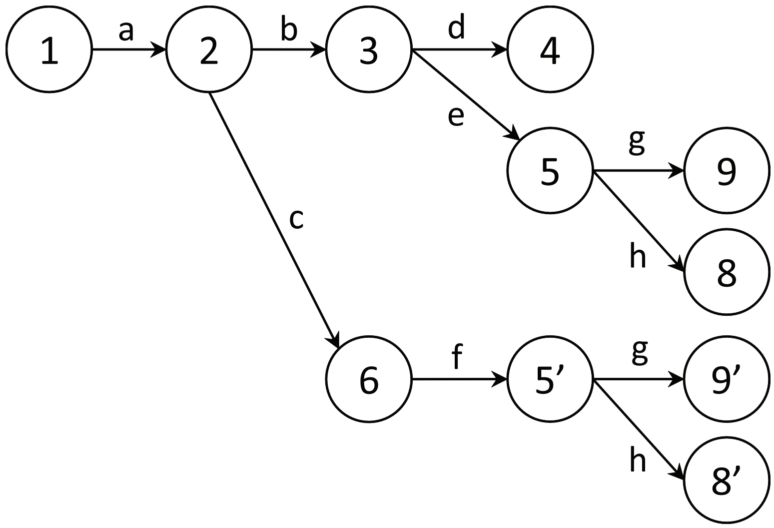

Figure 4: The branching story tree representation for the branching story graph in Figure 3

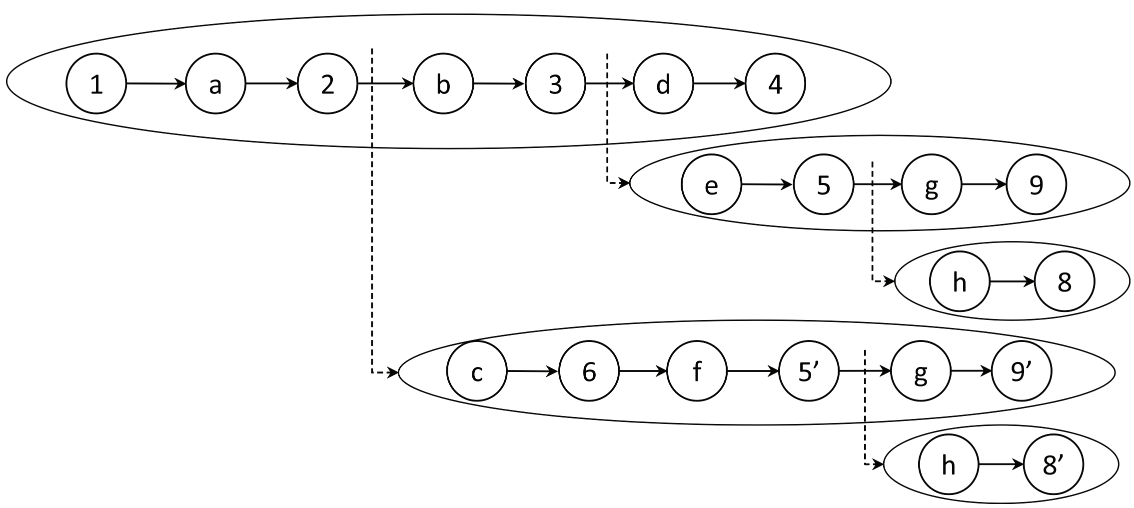

Figure 5: The narrative mediation tree for the branching story tree in Figure 4.

mediation system and the branching story graph [48, 73]. The narrative mediation system generates a linear narrative that represents the ideal story that should be told to the player and then considers all the ways in which the interactive player can interactive with the virtual world. For every action that the player makes that threatens to deviate severely from the linear story, the system generates an alternative storyline from the point of the deviation. Riedl and Young show that any acyclic branching story graph can be transformed into a branching story tree which can then be transformed into a mediation tree [49]. Figure 4 shows the branching story tree representation for the branching story graph in Figure 3. The tree is created by duplicating nodes with multiple incoming links. Figure 5 shows the narrative mediation tree for the branching story tree in Figure 4. For each branching point in the branching story tree, one of the branches is merged into a main linear story. All the other branches are converted into branches in the narrative mediation tree. On the other hand, any narrative mediation tree that does not contain concurrent player action and DM action can be converted into a branching story graph through converting all the player actions into branches of the branching story graph.

Search based DMs usually employ a *plot point graph* to move the interactive story forward. The plot point graph is a partially ordered graph where the nodes represent story event and the links represent temporal or logical constraints between the story events [39]. Unlike the branching story graph, the plot point graph does not literally specify all the possible stories. Any plot point and player action sequence that does not violate the constraints in the plot point graph is a legal story. Figure 6 shows a sample plot point graph used in Anchorhead [39]. A directed edge from node a to node b indicates that a must happen before b, unless multiple edges are joined with an OR arc. Figure 6 does not explicitly list all the possible stories trajectories as the branching story graph does. A legal story could start from the node *discover_safe*, followed by the node *get_crypt_key*, *get_safe_combo* and *open_safe*, etc.

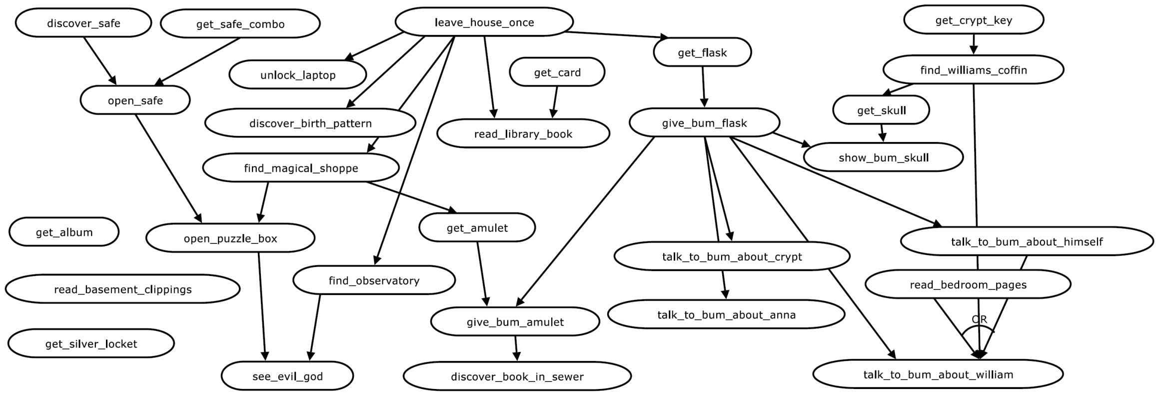

Figure 6: Plot point graph for Anchorhead's Day 2 [39]. A directed edge from a to b indicates that a must happen before b, unless multiple edges are joined with an OR arc.

An adversarial-like search algorithm can be used to find complete linear narratives in the space of possible narrative described by a plot point graph [70, 25, 49]. In theory, the adversarial-like search algorithm can be used in a breadth-first manner to find all the legal interactive stories which can then be transformed into a branching story graph, although the entire story space and the resulting branching story graph might be intractably large [25, 49].

With the branching story graph, the RQ1 can be characterized as the problem of modeling, at any given point, the players' preference on the successive plot points in the graph. Due to the sequential nature of stories, the preference over the successive plot points depends on all previous experienced plot points in the branching story graph. The prefix-based collaborative filtering algorithm is proposed to solve this problem in Chapter 3. RQ2 can be characterized as the problem of guiding the player to a selected plot point in the branching story graph. In Chapter 4, I describe a non-intrusive personalized guidance algorithm which can preserve the player agency. RQ3 will be characterized as the problem to select the a successive plot point that can achieve the highest expected preference ratings in the branching story graph. Chapter 4 will describe the personalized drama manager algorithm.

2.3 Drama Manager

The drama manager has been widely used in computer games to monitor the fictional world and guide players through an expected experience set by designers. Two most common types of drama managers include optimization-based (including search-based and declarative optimization-based) and planning-based drama manager. In this section, I will describe different types of drama mangers, followed by brief introduction of the player agency and how the drama mangers guide the players in previous interactive narrative systems.

2.3.1 Search-based and Declarative Optimization-based Drama Manager

The optimization-based drama manager, including search-based drama management (SBDM) and declarative optimization-based drama management (DODM), transforms the problem of selecting the next best plot point into a optimization problem where the DM chooses possible future histories of plot points by optimizing an objective function set by the designers.

The search based drama manager solves the optimization problem through searching for the best plot points in a pre-defined searching space [5, 70]. Most SBDMs describe the game as a plot point graph. The SBDM aims to search for the plot points and DM actions that can maximize an objective function while not violate the constraints in the plot point graph [54].

SBDM is first proposed by Bates [5] and developed by Weyhrauch [70]. Weyhrauch aims to optimize a dramatic aesthetic evaluation function that mimics the human artists in a simplified version of the Infocom interactive fiction Deadline, named Tea for Three. The evaluation function is calculated using predefined rules based on seven manually selected scenario features such as whether one event in the user's experience relates logically to the next, how bored the user feels, how much freedom the user perceives she has, etc. The output of the evaluation function is a real number between 0 (worst) and 10 (best), which represents the subjective aesthetic quality of the scenario. The system uses a variety of searching algorithms, such as full-depth adversary search, Sampling Adversary Search (SAS, SAS+), and memoized future contribution search (MMFC), to find plot points and DM actions that maximize the evaluation function. Weyhrauch finds that on "average" users, SAS, SAS+ and MMFC are able to improve users' experiences from the 50th percentile, to the 94th, 98th, and 99th percentile, respectively.

Nelson and Mateas further explore the SBDM and reproduce Weyhrauch's results [39]. They apply SBDM to an abstract story search space based on the text-based interactive fiction Anchorhead. Figure 7 shows a screenshot of the Anchorhead system. Each paragraph in the Figure is abstracted to be a plot point. Nelson and Mateas represent the story structure as a plot point graph as in Figure 6 and define five types of DM actions: permanent deniers, temporary deniers, causers, hints, and game endings. They define a story evaluation function based on seven manually selected features: location flow, thought flow, motivation, plot mixing, plot homing, choices, and manipulativity. They further use Weyhrauch's sampling search (SAS, SAS+ and MMFC) algorithms to find the DM actions that maximize the evaluation function. In their experiments, Nelson and Mateas found that Weyhrauch's excellent results with sampling search didn't transfer to their combination of plot points, evaluation function, and DM actions.

The declarative optimization-based drama manager is introduced by Nelson et al. [40]. In this work, the plot point abstraction, DM actions, player transition model, and author evaluation function are exactly as in [39]. But they approach the optimization problem through solving a Markov Decision Process (MDP) instead of searching. The author's evaluation function is used to define the reward function for MDP. The solution to the MDP represents the optimal choice of action for the DM given any history of plot points and DM actions. The solution of the MDP is a

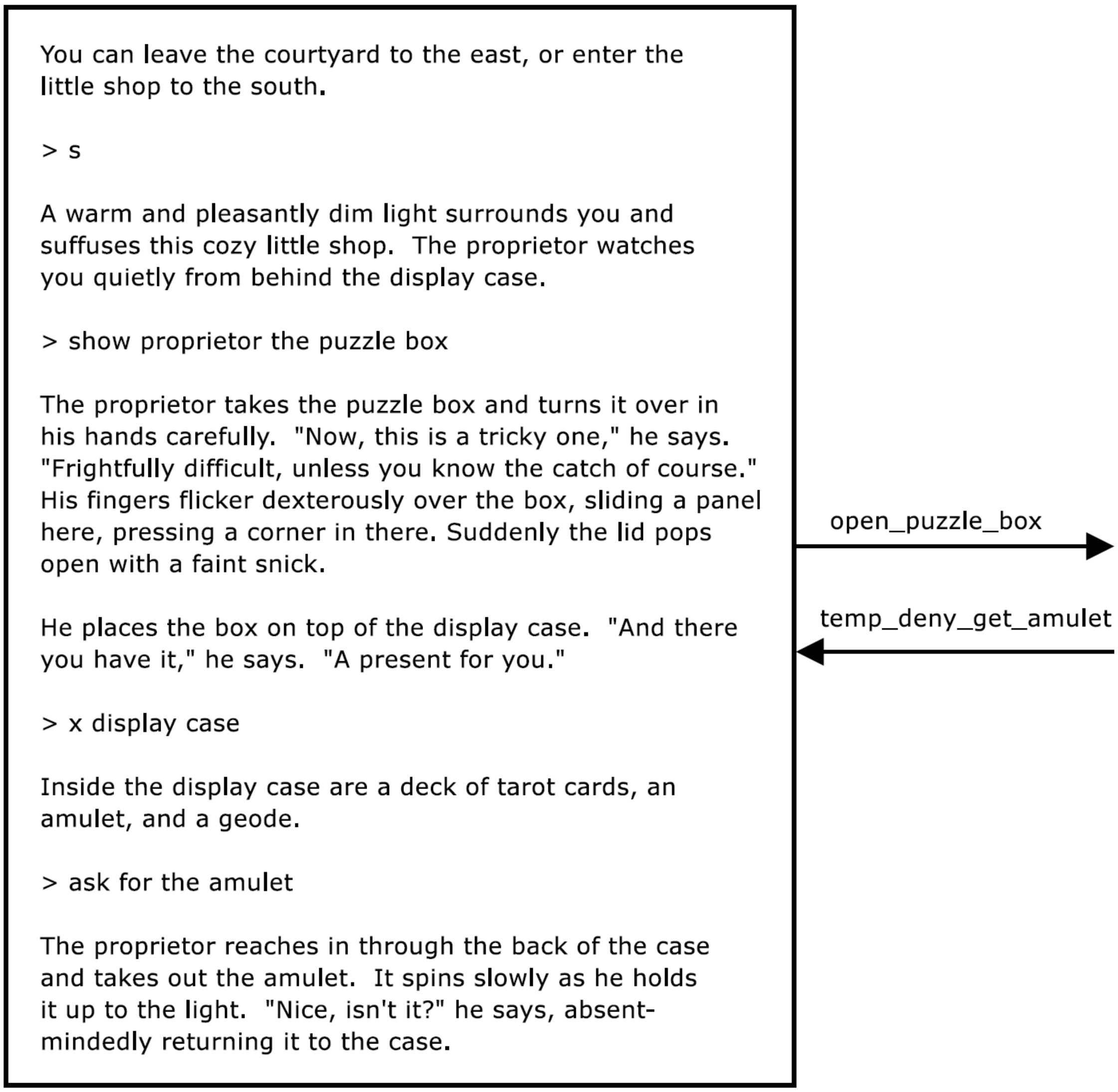

Figure 7: A screenshot from Anchorhead, showing the relationships between concrete game-world actions and the abstract plot points and DM actions. When the proprietor opens the puzzle box, the game recognizes this as the plot point open puzzle box and tells the drama manager. The drama manager decides on the DM action temp deny get amulet and sends it to the game, which implements it by not allowing the user to get the amulet.

policy indicating the optimal choice of action in every state that can maximize the long-term expected reward set by the designers. However they found that in the same Anchorhead model, RL does not do well either. They conclude that the DM actions specified are not sufficient to have much positive impact on the story, as measured by the given evaluation function—an authorship rather than optimization issue.

Targeted Trajectory Distribution Markov Decision Process (TTD-MDP), developed by Roberts, et al., is an alternative technique for solving the DODM problem [55, 6]. It solves non-Markov Decision Processes by wrapping all the previous MDP states into one node of a trajectory tree. A Targeted Trajectory Distribution MDP (TTD-MDP) is defined by a tuple (T, A, P, p), where A is a set of actions and P is a transition model the same as in traditional MDP. T is the set of finite-length trajectories (complete or incomplete) of MDP states which can also include actions. p is a target distribution over complete trajectories. The target distribution in a TTD-MDP conceptually replaces the reward function in a traditional MDP. The solution to a TTD-MDP is a policy $\pi : T \times A \rightarrow [0, 1]$ providing a distribution over actions for every trajectory. Their objective was to produce the probabilistic policy π that minimizes divergence from the target distribution of trajectories p. TTD-MDPs require a pre-defined target distribution across trajectories/stories, which needs to be specified beforehand. Further, as a reinforcement learning technique, it must simulate a player. While the simulated player may utilize a player model, that model would need to first be acquired or explicitly set by a human.

In summary, both the SBDM and the DODM aim to optimize an objective function or a reward function set by the human game designers. Thus they are all surrogates for the human designers. Furthermore, each objective or evaluation function is defined for a particular domain. It is not easy to transform the function across different domains.

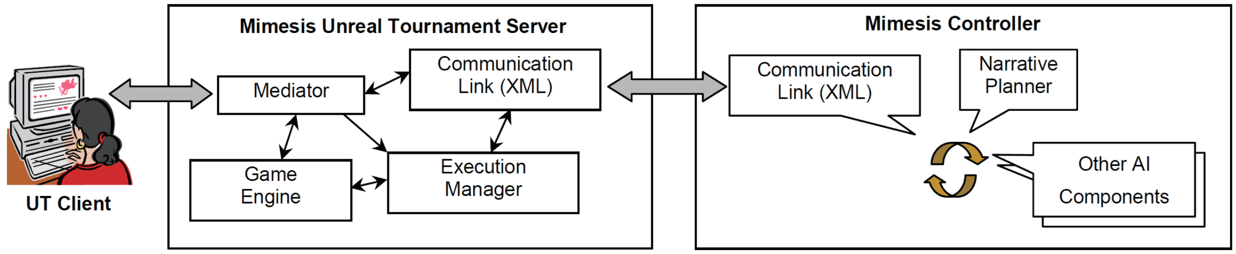

Figure 8: The Mimesis system architecture.

2.3.2 Planning-Based Drama Manager

Young and Riedl [72] develop a planning drama management system—Mimesis. Figure 8 shows the system architecture of Mimesis. Mimesis is consist of two components: the Mimesis Unreal Tournament Server (MUTS) and the Mimesis Controller (MC). The MUTS serves as an extended version of unreal tournament game engine. The MC acts as a story server, which is responsible for both the generation of a story and the maintenance of a coherent narrative experience in the face of unanticipated user activity. When a user performs actions that may interfere with the structure of the story plan, the Mimesis reactively responds to the actions by replanning or temporarily altering the effects of the players' actions to prevent the failure of the original narrative plans, which are specified by the game designers.

Riedl and colleagues [48, 73] propose narrative mediation which pre-computes every way the player can violate the original narrative plan and generates a contingency plan. It is applied in the Automated Story Director (ASD) [52], which uses a partial-order planner to re-plan a story when the player performs actions that change the virtual world in ways that prevent story progression as expected. Figure 9 shows a portion of pre-computed tree of narrative plan contingencies for the application of the ASD in Little Red Riding Hood.

Similarly, Porteous and Cavazza [45] use a planner with designer-provided constraints to control virtual characters and push a story forward. They formulate all the constraints the system needs to satisfy into a constraints tree (CT). Then the

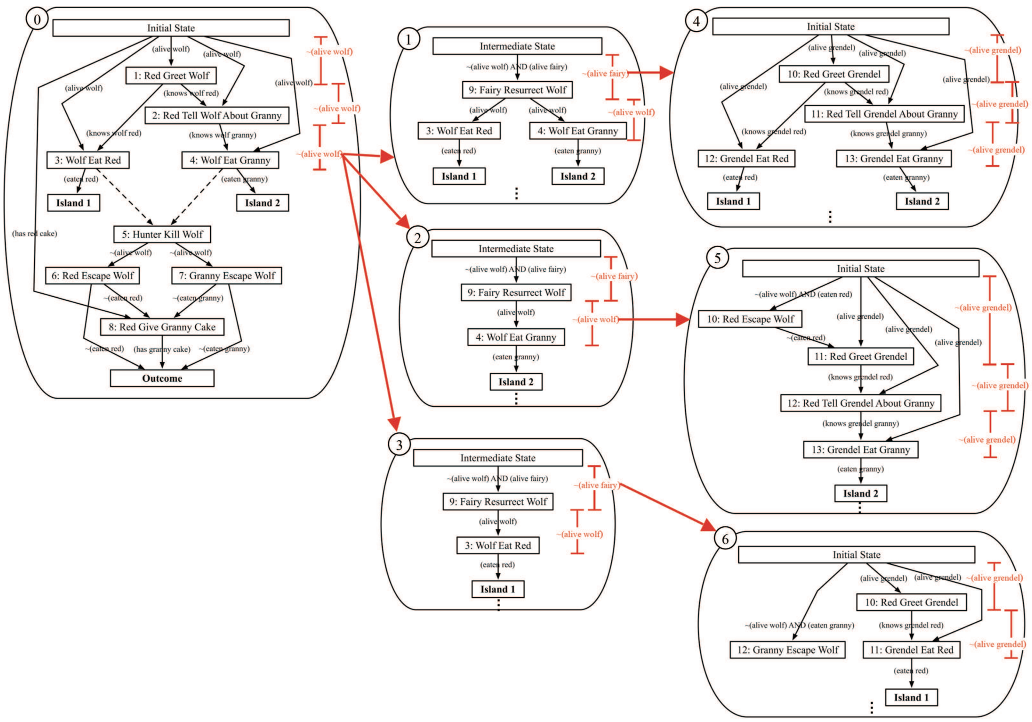

Figure 9: A portion of the tree of narrative plan contingencies for Little Red Riding Hood.

algorithm loops until the constraints tree is empty. Within each loop, the algorithm formulate a new subproblem, the goal of which is disjunctive and is formed from the leaves of the constraints tree, which are the earliest in the temporal order. After the subproblem is solved, the constraints tree CT is updated so that any facts in leaf nodes of the tree that are made true by the execution of previous plan are removed.

All of the above planning-based drama management techniques respond to player actions to move the story forward in a way partially or completely conceived by a human designer.

2.3.3 Other Types of Drama Manager

Other non-planning and non-optimization based drama managers use various kinds of techniques to choose plot points and DM actions. The Façade interactive drama [33] uses a reactive plot point selection technique to determine the next set of behaviors for

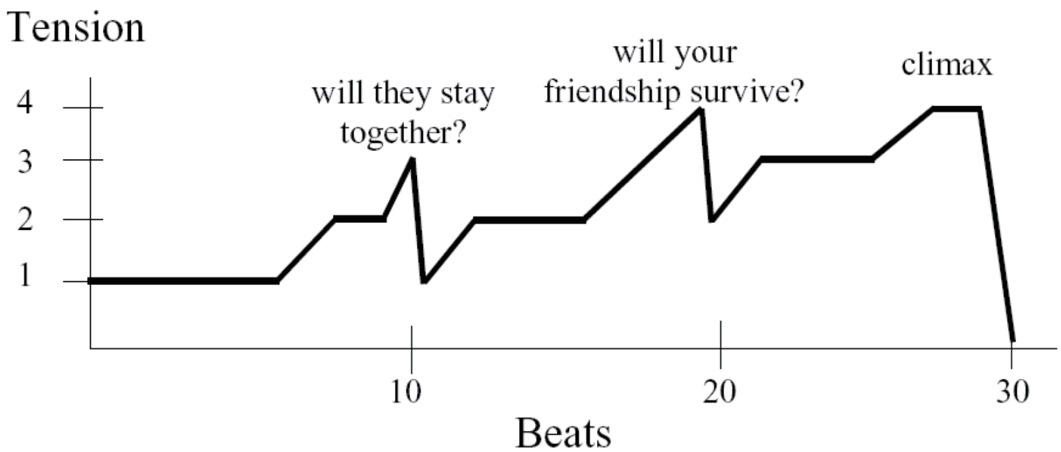

Figure 10: The Façade dramatic arc.

two virtual characters. It addresses the balance of character and plot by implementing a reactive behavior planner that selects, orders, and executes fine-grain plot elements called beats. A beat is the smallest unit of story structure that can move the story forward (e.g. a bit of dialogue or an action). Beats are selected and sequenced by a drama manager called a beat manager. The beat manager identifies beats that are applicable in the current state of the world and sequences the one that is most likely to achieve a particular story arc that is pre-defined by human designers. Figure 10 shows a dramatic tension arc for Façade.

The U-Director [36] uses dynamic decision networks, a generalization of Bayesian networks, to select DM actions to maximize narrative utility in the face of different uncertainties (player behavior). U-DIRECTOR explicitly models the uncertainty in narrative objectives, storyworld state, and user state. In each decision-making cycle, it systematically evaluates the available evidence, updates its beliefs, and selects the storyworld action that maximizes expected narrative utility. The narrative utility needs to be specified by designers and is similar as the objective function in Optimization-based drama manager.

Fairclough [13] proposes a narrative generation system OPIATE which uses case base (authored by game designers) and k-nearest neighbor to generate stories in response to the changing game environment and player actions.

These drama management systems do not take players' preference into consideration and are still surrogates for the human designers. In this research, I aim to create a personalized drama manager that is *also* a surrogate for the game players. The personalized drama manager takes into account both the designer's intention and the players' preference and preserves the player agency when it guides the players in the story space. In the next section, I will introduce player agency and briefly review how drama managers influence the players' choices in previous interactive narrative systems.

2.3.4 Player Agency and Player Guidance in Interactive Narrative

Player agency is an important aspect when evaluating the players' experience in interactive narrative and computer games. Murray [38] defines player agency as:

Definition 3 (Player Agency)**.** *The player agency is the satisfying power to take meaningful action and see the results of the decisions and choices.*

From Murray's definition, the player agency includes two important aspects: taking actions and see the results of the decisions. Nextly I will briefly review how the drama managers guide the players from the perspective of player agency in previous interactive narrative systems.

Nelson and Mateas [39] use five types of DM actions to to guide players in Anchorhead: permanent deniers which change the world so that a particular plot point becomes simply impossible for the duration of the game, temporary deniers which change the world so a particular plot point becomes temporarily impossible, causers which make a plot point happen, hints which make a plot point more likely to happen with an associated multiplier and duration, and game endings which are a special type of DM action that ends the game. Most of these DM actions are visible to the players. Some actions, such as permanent/temporary deniers, can even be intrusive for the players' experience. Some of the player actions are forbidden thus the player agency could be compromised in the system.

Magerko and Laird [31, 30] build an Interactive Drama Architecture (IDA) which uses planning and specific instructions such as "make the pistol jam" to actively prevent certain deviation of the storyline from the original goal. The Mimesis system and the narrative mediation also use a similar technique to intervene when the players perform actions that threaten the original story plan [72, 48, 73]. Although the players are not prevented from taking the actions, the results of their actions may not be what the players expected because of the intervention of the drama managers. Further studies are needed to evaluate the players' sense of agency in these drama management systems.

In this research, I create a personalized drama manager that does not forbid/hide any selection or explicitly changes the results of the players' action in the interactive narrative system. The personalized drama manager works in a *multi-option branching story graph*, which will be defined in Chapter 4, and guides the players through increasing the probability the players selecting the desired options. Furthermore, the personalized DM builds a player model to predict the players' preference over stories. There has been relatively little work that uses player models to determine how a story should unfold in a game or virtual environment. In the next section, I will briefly review previous work on player modeling and how the player models have been used to interactive narrative.

2.4 Player Modeling

Player modeling has been widely applied to adapt computer games in a variety of ways [29]. In this section, I will first briefly review related work on player modeling in general. Then I will discuss previous work on player modeling for interactive narrative systems.

2.4.1 Player Modeling in Computer Games

Player modeling has been widely used to dynamically adjust difficulty in computer games. Demasi and Cruz [10] employ genetic algorithms to build intelligent agents that best fit the players' skill level. They uses pre-defined models (agents with good genetic features) as parents in the genetic operations for their online coevolution. Hunicke and Chapman [19] control game environment settings to make the game easier or harder for different players. They employ a probabilistic method drawn from Inventory Theory for representing and reasoning about player status in first person shooter (FPS) games. Spronck, etc., [63] use dynamic scripting—an adaptive rule-base system—to adjust difficulty dynamically to the player skills. The rules in the rulebases are manually designed. The probability a rule is selected is based on an attached weight value which is updated to maintain a certain challenge level for players. Hartley and Mehdi [17] use cased-based approach to predict human players' actions and adapt the challenge level for FPS games. They develop a dual state representation to enhance case matching, and use adaptive k-d tree-based techniques to improve case storage and retrieval. Kazmi and Palmer [22] use finite state machine to recognize the skill level of players in FPS games. They try to make the game harder for expert players and easier for beginners. The finite state machine is also used to adapt NPC behaviors, weapon mechanics and game level geometry in their research. Collaborative filtering can also be used in the player modeling problem [34]. Zook and Riedl [81] use tensor factorization to learn player skill mastery over time, allowing more accurate tailoring of challenges. They demonstrate the efficacy and scalability of tensor factorization models in a simple role-playing combat game.

Player modeling has also been applied in procedural content generation to generate the game world, scenarios and quest for different players. Magerko et al. [32]

build an intelligent director agent to customize simulation training scenarios for different trainee. They build a skill model that captures player proficiency levels by monitoring and rating the trainee's actions. The plot points in the training scenarios are selected by matching the tested skills with the skill model of the trainees. Togelius, etc, [69] propose a neural network based player modeling approach to infer and simulate players' behaviors and predict entertainment levels of the players. Their approach can dynamically generate tracks to increase player satisfaction in a car racing game. On a similar direction, Shaker, etc. [60] also use the neural network based player modeling approach to adapt level design parameters for different types of players in platform games. They use exhaustive search to find parameters for the neural network to maximize players' satisfaction. Jennings-Teats et al. [20] propose Polymorph, a dynamic level generation system that can adjust difficulty levels for 2D platform games. They use Rank-SVM to learn game level models and player models from human labeled data. Sullivan et al. [65] propose the Grail Game Manager, a rule-based system which can dynamically generate personalized quest structures for different players. They choose quest entities (goals, actions, rewards, NPCs, and dialog options) using players' history and current world state.

2.4.2 Player Modeling in Interactive Narrative

The Interactive Drama Architecture (IDA) [31, 30] predicts player actions and reasons about potential threats using a predictive player model. Then it attempts to preserve the narrative goals—specified by the game designers—by directing virtual characters to perform actions or change goals. In the IDA, the player model is used to model players' actions instead of players' story preference. Thus it is still a surrogate for the game designers.

Relatively little work has been done to use player models to model player preference and determine how a story should unfold in a game or virtual environment. The

PaSSAGE system [67] automatically learns a model of the player's preference through observations of the player in the virtual world, and uses the model to dynamically select the branches of a CYOA style story graph. PaSSAGE uses Robin's Laws five game player schemes: Fighters, Power Gamers, Tacticians, Storyteller, and Method Actors. A player is modeled as a vector where each dimension is the strength of one of the types. As the player performs actions, dimensions are increased or decreased in accordance to rules.

Peinado and Gervás [41] build player models using the same player types as in the PaSSAGE system. They use a knowledge intensive case based reasoning approach to generate interactive stories based on the current game state and the player models.

Seif El-Nasr [12] propose Mirage which uses a four-dimension player model: heroism, violence, self-interestedness, and cowardice. Mirage uses a pre-authored rule-based system to associate player behaviors to the four dimensional player models then selects narrative events based on author's specification and the player models.

Sharma etc. [61] use case based reasoning player modeling approach to predict interestingness values for the plot points within the story. The drama manager then uses an expectimax search process based on the player model and the author specified aesthetic values to select plot points for interactive narrative systems.

These player modeling techniques assume players can be classified according to several discrete play styles and that, even with continuous characteristic vector combining the discrete player styles, optimal story choices can be made by a DM (Sharma's DM is an exception. But their player models are also built based on a fixed set of hand-chosen features) [74, 77]. These systems further assume that role playing game player classifications (or ad-hoc types) are applicable to story plot choices. In addition, these systems assume that plot points could be selected in isolation from each other based on a comparison between their attributes and the player model. In this research, I propose a collaborative filtering based player modeling approach that learns player model dimensions from player feedback—ratings—and further solves sequential plot point recommendation/selection problems. In the next section, I will introduce some background on collaborative filtering and its application in computer games.

2.5 Collaborative Filtering and Its Application in Computer Games

In this section, I will firstly review various approaches for collaborative filtering. Then I will discuss its applications in computer games.

2.5.1 Collaborative Filtering

As one of the most successful approaches to building recommender systems, collaborative filtering make use of known preferences of a group of users to make recommendations or predictions of the unknown preferences for other users. Collaborative filtering algorithms can be categorized into memory-based collaborative filtering, model-based collaborative filtering, and hybrid collaborative filtering [1].

Memory-based collaborative filtering algorithms use the entire or a sample of the user-item database to model the user preference. Neighborhood-based CF and item-based/user-based top-N recommendations are typical examples of memory based CF [58]. The neighborhood-based CF calculates the similarity or weight $w_{i,j}$, which reflects distance, correlation, or weight, between two users or two items, i and j. Then it produces a prediction for the active user by taking the weighted average of all the ratings of the user or item on a certain item or user, or using a simple weighted average [58]. The memory-based collaborative filtering is an early approach used in many commercial systems. It is easy to implement and fast to compute. But its performance decreases when data gets sparse.

[1] http://en.wikipedia.org/wiki/Collaborative˙filtering

Model-based collaborative filtering algorithms use data mining and machine learning algorithms to find patterns based on training data. The learned models are used to make predictions for new users. Prevailing model-based collaborative filtering algorithms include Bayesian networks, clustering models, latent semantic models such as singular value decomposition and non-negative matrix decomposition, probabilistic latent semantic analysis such as probabilistic principal component analysis, multiple multiplicative factor, latent dirichlet allocation and markov decision process based models [64]. The model-based collaborative filtering algorithms handle the sparsity better than memory-based ones. Thus they improve the prediction performance with large data sets. The model-based CF algorithms usually give an intuitive rationale for the recommendations, although it can be time consuming to train the models.

The hybrid collaborative filtering algorithms combine the memory-based and the model-based CF algorithms. The hybrid approaches try to overcome the limitations of both memory-based and model-based algorithms. However, they usually increase complexity and are expensive to implement [15].

In this research, I compare and use various memory-based and model-based collaborative filtering algorithms, including K-means clustering, K-nearest neighbors, probabilistic principal component analysis, and non-negative matrix factorization, etc., to build the player preference models. In the next section, I will discuss previous applications of collaborative filtering for the player modeling in computer games.

2.5.2 Application in Computer Games

Collaborative filtering has not been widely used in computer games to model the player preference or build the player models. Melder suggests that game developers can benefit from incorporating the recommendation algorithms to build player models and adapt gameplay based on the player preference [34]. Zook and Riedl [81] use a tensor factorization approach to predicting player performance in skill-based computer games. The tensor factorization approach is a temporal based collaborative filtering that can predict changes in players' skill mastery over time, allowing more accurate tailoring of challenges.

Most common collaborative filtering algorithms, including the temporal based collaborative filtering, cannot be used directly to model the players' preference over story plot points due to the sequential natural of stories. The prefix based collaborative filtering algorithm I develop in this research address the sequential recommendation problem and can be used to predict the players' story preference.

CHAPTER III

DATA-DRIVEN PLAYER MODELING ALGORITHM

In this chapter, I will describe a data-driven player modeling algorithm—prefix-based collaborative filtering (PBCF)—that predicts players' preference over successive plot points and chooses successive plot points for the players in storytelling systems.

Previous approaches to personalization of story generation and interactive narrative [67, 41, 11] require a designer to pre-determine the meaningful player types, even though there is no comprehensive theory that these pre-defined player types can cover all different players or can be generalized to different types of games, nor any clear evidence of links between player selections and the pre-defined player types. In this research, I use collaborative filtering (CF) to discover the player types directly from data. Collaborative filtering has been widely used in recommender systems to model user preference over movies, books, music, and other products from users' structural feedback, e.g. previous ratings, historical purchase, etc [64]. The CF algorithms are data-driven approaches that attempt to learn users' preference patterns from ratings and predict new user's ratings from previous users' ratings which share similar preference patterns. The CF algorithms enable us to easily extract the player types by observations of players' structured feedback, i.e. ratings. The player models built with CF algorithms do not take any assumption about pre-defined player types.

Unlike the traditional usage of CF for *one-shot* recommendations of complete story artifacts (e.g., books, movies), modeling preference over successive plot points is a new type of recommendation problem—*sequential recommendation*, in which each subsequent recommendation is dependent on the entire sequence of prior recommendations for a particular story experience. In other words, the sequential recommendation is a

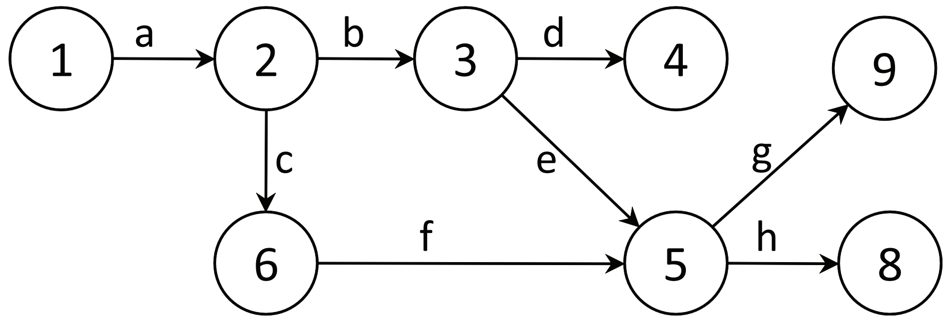

Figure 11: A sample branching story graph.

non-Markovian problem—at each step the DM's selection of best next plot point is based on all previous plot points encountered by the player. Thus most traditional CF algorithms cannot be used directly to model the player's preference over successive plot points. For example in Figure 11, which is the same example as Figure 3 in Chapter 1, if a player is currently at node 5, the DM's next selection, node 8 or node 9, should depend on the player's previous experience, i.e., {1, 2, 3, 5} or {1, 2, 6, 5}, instead of preference on the individual node 8 or node 9. The player's preference on {1, 2, 3, 5, 9} and {1, 2, 6, 5, 9} may be completely different. The PBCF algorithm is developed to address the sequential recommendation problem.

The prefix-based collaborative filtering algorithm first converts the branching story graph into a prefix tree representation. Then it models the players' preference over the prefix nodes and select successive plot points based on the players' preference over the prefix nodes. Section 3.1 and Section 3.2 describe the PBCF algorithm in more details. Section 3.3 evaluates the PBCF algorithm on both human data and simulated data in a story generation system I built based on choose-your-own-adventure books.

3.1 Prefix Tree Representation

The first step to address the sequential recommendation problem is to transform the branching story graph into a *prefix tree*. Figure 12 shows a prefix tree transformed from Figure 11. In Figure 12, every node is a prefix of a possible complete story (i.e.,

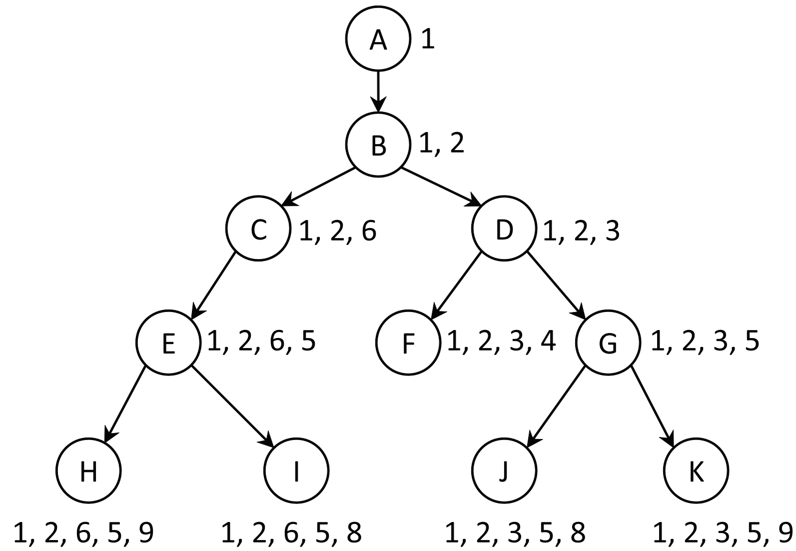

Figure 12: The corresponding prefix tree of the branching story graph in Figure 3. Options are temporarily ignored in this figure. The branching story graph is usually a graph while the prefix tree is a tree or forest.

a path from initial node to terminal node in a branching story graph). The children of a node in the prefix tree are the prefixes that can directly follow the parent prefix. Comparing Figures 11 and Figure 12, one can see that each node in the prefix tree incorporates all the previous plot points in the path from the initial plot point to the current plot point. With the prefix tree, the drama manager does not need to worry about history when choosing the next node because the previous history is incorporated into the prefix nodes themselves.

In my approach, stories will be presented to the players plot point by plot point and I will collect players' ratings for the "story-so-far", the portion of the story that they have observed leading up to the current point in time. Notice that it is easier and more accurate for the players to rate the story-so-far than the new plot point only since the history of interaction up to that point matters in stories. Any one plot point does not make sense without previous ones. It is probably impossible to ask the players to not consider the context of the prior story. The story-so-far exactly corresponds to the prefix nodes in the prefix tree. Furthermore, through prefix tree I do not need to solve the credit assignment problem as in reinforcement learning which is to determine how much of a final rating each previous plot point is responsible for [66].

Compared to other algorithms which also roll history into state nodes such as TTD-MDP [55, 6], the prefix based collaborative filtering algorithm focuses on optimizing the path for different players based on the player model. Furthermore, unlike TTD-MDP which tries to convert the problem into a Markovian problem, the prefix selection problem in the paper is still non-Markovian. For example, if the DM is at node D in Figure 12, the selection of the next node (F or G in Figure 12 should be related to the player's ratings on previous three prefix nodes (A, B, and D). A player who gives positive feedback on node B and negative feedback on node D should be different from another one who gives negative feedback on both node B and D. Through the prefix based CF algorithm I describe in the next sections, the DM can model players' preference based on all previous prefix ratings and make a selection to the best of its knowledge.

3.2 Prefix Based Collaborative Filtering

With the prefix tree representation, a *prefix-based* collaborative filtering algorithm can now be considered to build the player preference model. In prefix based collaborative filtering, all the players' ratings for the story prefixes are collected in a single matrix which I call a *prefix-rating matrix*. An n by m prefix-rating matrix contains the ratings for n prefixes from m players. Each column of the matrix represents the ratings of the corresponding player for the all the prefixes. Each row of the matrix represents ratings for the corresponding prefix from all the players. Figure 13 shows a simple illustration of the prefix-rating matrix. The matrix is usually very sparse, i.e. containing a lot of missing ratings which are labeled as $*$ in Figure 13, because I do not expect any given player to have read and rated all the prefixes in the library.

Prefix	Player 1	Player 2	Player 3	...
A (1)	*	*	2	...
B (1, 2)	1	*	2	...
C (1, 2, 6)	*	*	*	...
D (1, 2, 3)	4	3	*	...
...	...	...	...	...

Figure 13: An illustration of the prefix-rating matrix. A, B, C and D represent the prefixes in Figure 12. The larger the digital number, the higher the preference. The stars represent those missing ratings.

If one can predict all the missing ratings in the prefix-rating matrix for a player, it will be straightforward to choose the best next prefix during story generation process—to choose the prefix that will lead to the highest rated full-length stories for the player. In my approach, the prefix-rating matrix is treated as the product-rating matrix as in traditional Collaborative Filtering [64, 79]. CF algorithms can be applied to train and compute the missing ratings in the prefix-rating matrix.

The CF algorithms make no presumptions on the player types as in [67]. Instead the algorithms will cluster the ratings of the players and learn "patterns" from each cluster of ratings. These "patterns" also represent player types, although, as with many machine learning techniques, it is difficult to interpret these player types. These player types are soft clusters in the sense that a particular player can have a degree of membership in each player type. The learned player types is more capable of describing all types of players for particular games, compared to the pre-defined types.

The PBCF algorithm consists of two phases: model learning and story recommendation. In the next section, I will introduce the CF algorithms that will be used in this work. Then I will describe two phases in the PBCF process: model learning and story recommendation.

3.2.1 Collaborative Filtering Algorithms

I experimented on two collaborative filtering algorithms: *probabilistic Principal Component Analysis* (pPCA) [68] and *Non-negative Matrix Factorization* (NMF) [26, 80]. I will briefly introduce the two algorithms and their application to the model learning and story recommendation process.

The probabilistic PCA algorithm assumes that a n dimensional vector $\mathbf{r}$ can be factorized as follows:

$$\mathbf{r} = W\mathbf{x} + \boldsymbol{\mu} + \boldsymbol{\epsilon} \tag{1}$$

where $\mathbf{x}$ is a n' dimensional vector in the hidden or reduced dimension space (usually $n' < n$) and $\mathbf{x} \sim N(0, \mathrm{I})$. W is a n by n' matrix. $\boldsymbol{\mu}$ is the mean vector which permits $\mathbf{r}$ to have nonzero mean. $\boldsymbol{\epsilon} \sim \sigma^2\mathrm{I}$ is the Gaussian noise.

Let the vector $\mathbf{r}$ be one column of the prefix-rating matrix. pPCA projects the corresponding player's prefix-rating vector into a hidden space or a reduced dimension space $\mathbf{x}$ just as in traditional principal component analysis. The hidden space vector $\mathbf{x}$ models the corresponding player's preference type. Note that from Equation 1:

$$\mathbf{r}|\mathbf{x} \sim N(W\mathbf{x} + \boldsymbol{\mu}, \sigma^2\mathrm{I}) \tag{2}$$

Thus the basic assumption of pPCA algorithm is that the player's prefix rating vector (the column of the prefix-rating matrix) obeys a multi-dimensional Gaussian distribution. In other words, the ratings for each prefix from a single player take a univariate Gaussian distribution. Furthermore, pPCA assumes that the expectations of different players' rating vectors are linear combinations of w_i, the columns of the matrix W, which in my case represent player types. The player model in pPCA is captured by W, μ and σ. Thus, for each player with prefix-rating vector $\mathbf{r}_i$, these parameters help us find the hidden vector $\mathbf{x}_i$, the individual player's preference properties; once the hidden vector is known, the player's ratings for all prefixes without ratings by this individual can be computed according to the multi-dimensional

Gaussian distribution.

The NMF algorithm aims to factorize an n by m matrix R as follows:

$$R = W * H \tag{3}$$

where $W \in \mathbb{R}^{n*n'}$ and $H \in \mathbb{R}^{n'*m}$ are two non-negative matrices (usually $n' < n$). The non-negative property means that all the entries in the matrix are greater than or equal to zero.

In my case, R is set to be the prefix-rating matrix (n prefixes and m players). The player model in NMF is simply the matrix W. The columns of the matrix W, $\boldsymbol{w}^j$ $j = 1, ...n'$, are bases that represent different types of players. $\boldsymbol{h}^i$, the i^{th} column of H, corresponds to the i^{th} player's preference properties.

In practice, it will be difficult to interpret the player types that correspond to w_i or $\boldsymbol{h}^i$. However, known player types can be introduced as prior knowledge to the model learning process to improve the training accuracy. For example, in NMF, if I have prior knowledge about some preference types (e.g., fighter, tactician), i.e., I know their ratings for all the prefixes (e.g., fighter's rating vector $\boldsymbol{w}^f$, tactician's rating vector $\boldsymbol{w}^t$), then the matrix W can be seeded with the rating vectors ($\boldsymbol{w}^f$, $\boldsymbol{w}^t$) as fixed columns.

3.2.2 Model Learning Algorithms

Due to the large amount of missing values in the prefix-rating matrix R, EM algorithm [2] is used to learn the parameters for pPCA algorithm (W, $\boldsymbol{\mu}$ and σ) and NMF algorithm (W).

3.2.2.1 Model Learning with pPCA

In the E-step of the pPCA model learning algorithm, I use a Gaussian Process to compute the missing ratings in R given the parameter W, $\boldsymbol{\mu}$ and σ [47]. Let $\Sigma = WW^T + \sigma^2 I$ be the covariance matrix for the rating vector $\mathbf{r}$, which is one column of R. Denote the sub-vector of $\mathbf{r}$ which contains all missing ratings as $\mathbf{r}_h$ and the sub-vector of $\mathbf{r}$ which contains all known ratings as $\mathbf{r}_o$. Then the distribution $p(\mathbf{r}_h|\mathbf{r}_o)$ and the expectation of $\mathbf{r}_h$ can be computed using the Gaussian Process:

$$\mathrm{E}(\mathbf{r}_h|\mathbf{r}_o, \boldsymbol{\mu}, \Sigma) = \boldsymbol{\mu}_h + \Sigma_{ho}(\Sigma_{oo})^{-1}(\mathbf{r}_o - \boldsymbol{\mu}_o) \tag{4}$$

where $\boldsymbol{\mu}_h$ is the sub-vector of $\boldsymbol{\mu}$ containing elements at the positions corresponding to the missing ratings and $\boldsymbol{\mu}_o$ is the sub-vector containing elements at the positions corresponding to the known ratings. Σ_{ho} means the sub-matrix of Σ, of which the rows are indexed by the position of missing ratings while the columns are indexed by known ratings in $\mathbf{r}$. The notation system follows the tradition in [79].

In the M-step of the pPCA, the parameters W, $\boldsymbol{\mu}$ and σ are computed through maximizing the expected likelihood function $\mathrm{E}(\mathbf{r}_h, \mathbf{r}_o|W, \boldsymbol{\mu}, \sigma)$ over distribution $p(\mathbf{r}_h|\mathbf{r}_o)$, which is computed in the E-step using Gaussian Process. After a few equation manipulations, the expected likelihood function will be:

$$\mathrm{E}(\mathbf{r}_h, \mathbf{r}_o|W, \boldsymbol{\mu}, \sigma) \sim \log|\Sigma| + \mathrm{tr}(C\Sigma^{-1}) \tag{5}$$

where $C = \frac{1}{m}\mathrm{E}(\sum_{i=1}^{m}(\mathbf{r}_i - \boldsymbol{\mu})(\mathbf{r}_i - \boldsymbol{\mu})^T)$ and m is the total number of training players. The parameters can be computed by minimizing Equation 5. The minimization results are as follows:

$$\boldsymbol{\mu} = \frac{1}{m}\sum_i \mathbf{r}_i \tag{6}$$

$$\sigma^2 = \frac{\mathrm{tr}(C) - \sum_{i=1}^{n'}\lambda_i}{n - n'} \tag{7}$$

$$W = U'S \tag{8}$$

where the λ_i is the ith biggest eigenvalue of C, U' contains the n' eigenvectors corresponding to λ_i and S is a diagonal matrix with the i value equaling to $(\lambda_i - \sigma^2)$.

Initialize W, $\boldsymbol{\mu}$ and σ randomly for pPCA, or W and H for NMF
while not converging or termination criterion not reached **do**
    Compute the rating matrix R through Equation 4 for pPCA algorithm, or Equation 3 for NMF algorithm ▷ E-step
    Set the corresponding elements in R to the known ratings in R_0
    Compute W, $\boldsymbol{\mu}$ and σ through Equation 6-8 for pPCA, or W and H through Equation 9 and Equation 10 for NMF ▷ M-step
end while

Figure 14: The model learning algorithm.

3.2.2.2 Model Learning with NMF

For the NMF algorithm, the E-step to compute the prefix-rating matrix R is simple given the W and H: $R = WH$. Then the known elements of R are set to be the corresponding input player ratings.

Given a fully observed R in the M-step, the objective is to minimize the distance $||R-WH||^2$ as in [26] to get W and H, where $||\circ||$ is the Frobenius norm. The update rules are as follows:

$$H_{ij} \leftarrow H_{ij} \frac{(W^T R)_{ij}}{(W^T W H)_{ij}} \tag{9}$$

$$W_{ij} \leftarrow W_{ij} \frac{(R H^T)_{ij}}{(W H H^T)_{ij}} \tag{10}$$

3.2.2.3 Summary of Model Learning

Regardless of algorithm, the complete model learning process is as follows. In the first step, I build the story library and convert the branching story graphs into a group of prefix trees. In the second step, I collect data to populate the prefix-rating matrix R_0. In the third step, I use the algorithm in Figure 14 to learn the player model, i.e., to predict missing rating values for all players.

3.2.3 Story Recommendation Algorithms

At current stage, the players are not allowed to select options by themselves. Instead, the DM will recommend a path through the branching story graph for an individual player based on the learned player models in the model learning phase. The story

```
Initialize h, a column of H in Equation 3, for NMF
while Not converging or termination criterion not reaching do
    Compute r using Equation 3 (with h)
    Set the corresponding elements in r to the known ratings in r_0
    Compute new h using Equation 9
end while
```

Figure 15: The rating prediction algorithm for NMF.

recommendation phase begins with collecting some initial prefix ratings to seed $\boldsymbol{r}_0$ for a new player. Next, I predict the missing ratings for the player using the player models built in the modeling learning step. When using pPCA, this is accomplished simply by applying Equation 4. For NMF algorithm, the prediction algorithm in Figure 15 is used. At last, the plot point in the prefix that can lead to the highest rated full-length story will be selected.

For example if the story has proceeded to node B in Figure 12, the selection of plot point C or D depends on the predicted ratings of node H, I, J and K. If node I gets the highest predicted rating, then the DM will select plot point 6. If node K wins, plot point 3 in node D will be selected.

3.2.3.1 Summary of Story Recommendation

The entire story recommendation phase can be described as follows:

1. Collect a initial rating vector $\boldsymbol{r}_0$ with missing values from the player.
2. Compute $\boldsymbol{r}$ with no missing values using Equation 4 for pPCA, or the algorithm in Figure 15 for NMF.
3. Select one of the child nodes in the prefix tree that can lead to the highest rated leaf node in the prefix tree.
4. Present the plot point from the selected prefix node that would follow the current plot point.
5. Collect player's rating on the story-so-far (i.e., the recommended prefix).

6. Include the new rating into $\boldsymbol{r}$ and go to step 2.

Notice that it is not necessary to collect new ratings after each prefix in the story recommendation phase; I do it in the system for the purpose of collecting as much data as possible to build a more accurate player model. With every new rating, the DM will get better understanding of the current player's preference. This is important as the quality of the recommendation can improve and prior predictions can be overridden. The DM can change the recommendation direction as soon as it finds the prior predictions are wrong. In fact, I show in the experiments with simulated players that the prefix based story generation approach works better than applying collaborative filtering directly on full-length stories.

3.3 Evaluation of the Player Modeling Algorithm

In this section I describe human studies and simulated studies designed to evaluate the prefix based collaborative filtering algorithm in a simple story generation environment. I will first introduce the story library used in the experiment, followed by the user interface of the story generation system. Then I will describe four human studies and one simulated study I performed. The first human study trained a player model on human ratings of stories. I evaluate pPCA and NMF implementations for the accuracy of models of human rating behavior learned. The second human study used the player model trained in the first experiment to evaluate the story recommendation algorithm against a baseline. The third and the fourth human studies are comparison experiments which evaluate story preference model built without considering history and story preference model built with the Robin's Laws as prior knowledge. Finally, I will describe experiments conducted on simulated computer players in order to get a more complete picture of PBCF algorithm performance.

3.3.1 Story Library

The story library I used for the experiments was built by transcribing the stories from four Choose-Your-Own-Adventure (CYOA) books: *The Abominable Snowman*, *Journey Under the Sea*, *Space and Beyond*, and *The Lost Jewels of Nabooti*, all of which contain adventure stories for teenagers in the US [1]. All the stories from one book constitute a branching story graph. At the end of every page in the book, there is a multi-choice question. Depending on which choice the reader chooses, he or she will be delivered to different pages of the book to continue down different branches of the story. Figure 16, which is the same as Figure 1 in Chapter 1, shows a branching story graph from one of the books—*The Abominable Snowman*. Here I do not allow the players to select options by themselves for the purpose of evaluating the story preference models.

One of the reasons that I transcribed stories from Choose-Your-Own-Adventure books is to control for story quality, as opposed to authoring stories myself. In the current system, every story was pruned and transcribed to contain exactly six plot points. This was achieved by manually removing branches that led to "sudden death" outcomes and merging a few successive plot points. Note that the algorithm generalizes to longer and variable-length stories, although all the stories are of the same length in the current library. Figure 17 shows the branching story graph of *The Abominable Snowman* after conversion.

The four branching story graphs were transformed into four prefix trees (in this case a prefix forest), which was stored in the story library. In total, the story library is capable of generating 154 possible stories (about 1000 words per story) and contains 326 prefixes. The constraints between plot points greatly reduce the number of valid stories in the story library such that the number of prefixes will grow linearly with the number of full-length stories.

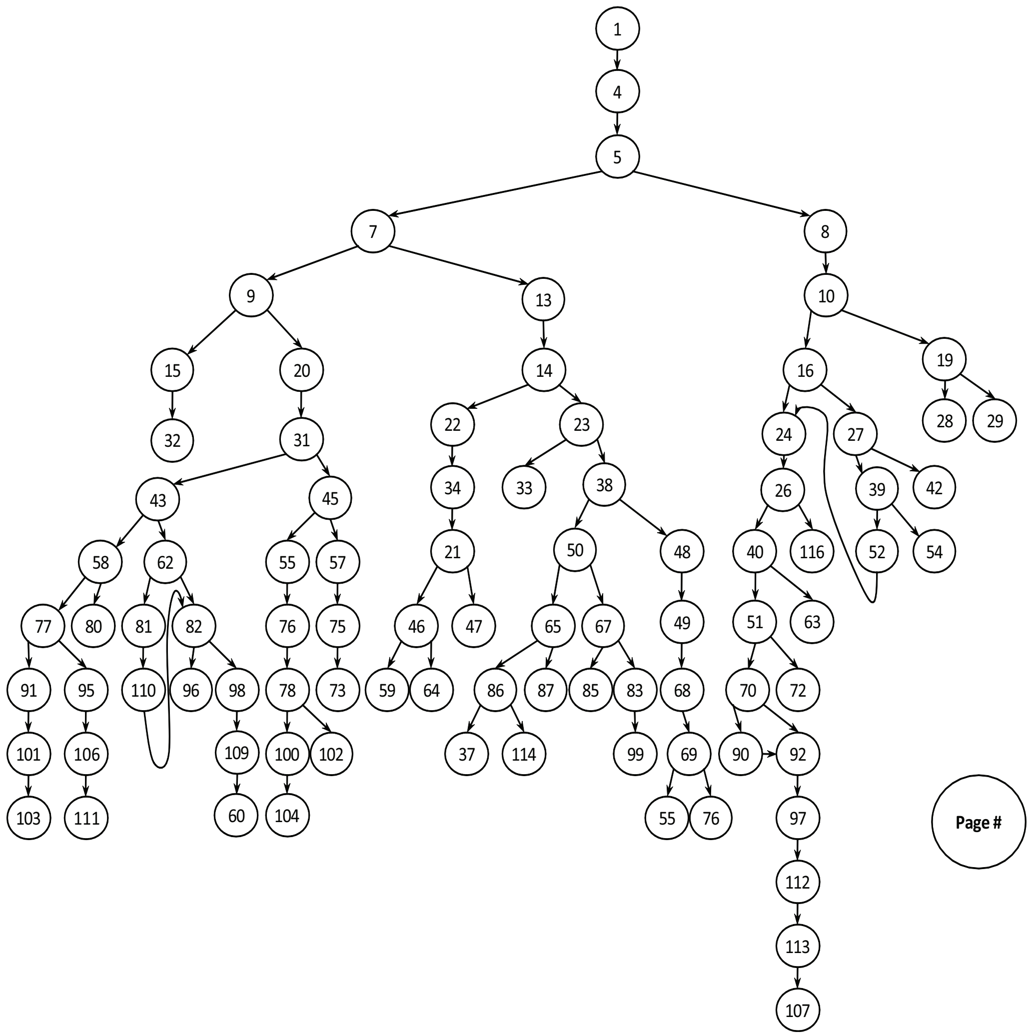

Figure 16: Illustration of the branching story graph of stories in *The Abominable Snowman*. Each node in the graph represents a page in the book (a plot point). Every story starts from the root node and ends on one of the leaves.

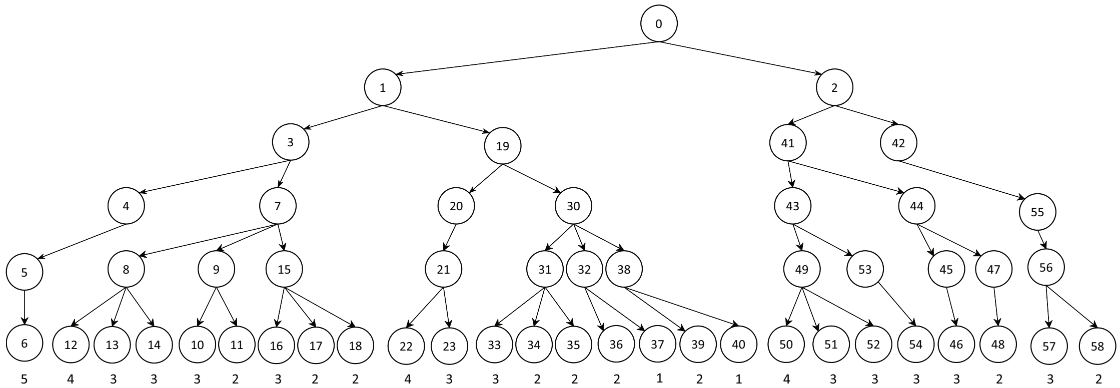

Figure 17: The branching story graph for the choose-your-own-adventure book: *The Abominable Snowman*. The digits at the bottom are the left-most score distribution I used in the evaluation.

3.3.2 User Interface

I built a simple story generation system to evaluate the prefix based CF algorithm. The system implements the CF model learning and story recommendation algorithms described earlier. It selects the next plot point that is believed to be most enjoyed by the player then directly present the plot point to the player.

Figure 18 shows a screenshot of the simple story generation system I built for experiments with human participants[1]. The system presents the stories to the player one plot point at a time, where each plot point is a paragraph of a story. In Figure 18, two plot points are presented. After each plot point, the system asks the player for their preference rating on the story-so-far (corresponding to a prefix node in the prefix tree). The ratings are integers ranging from 1 to 5, where a larger number indicates a higher preference. A new plot point will appear after the player clicks the *Next* button. The next plot point is determined by the story selection algorithm, which is either PBCF or random selection of a successor. In this system, by limiting player interaction to providing ratings of the story-so-far I aim to control the experimental

[1]In this human study, I collected the players' preference ratings using a slider, which could introduce a default bias effect [18]. In the next chapter, I will use a group of radio buttons without pre-selecting any of them to avoid the default bias effect.

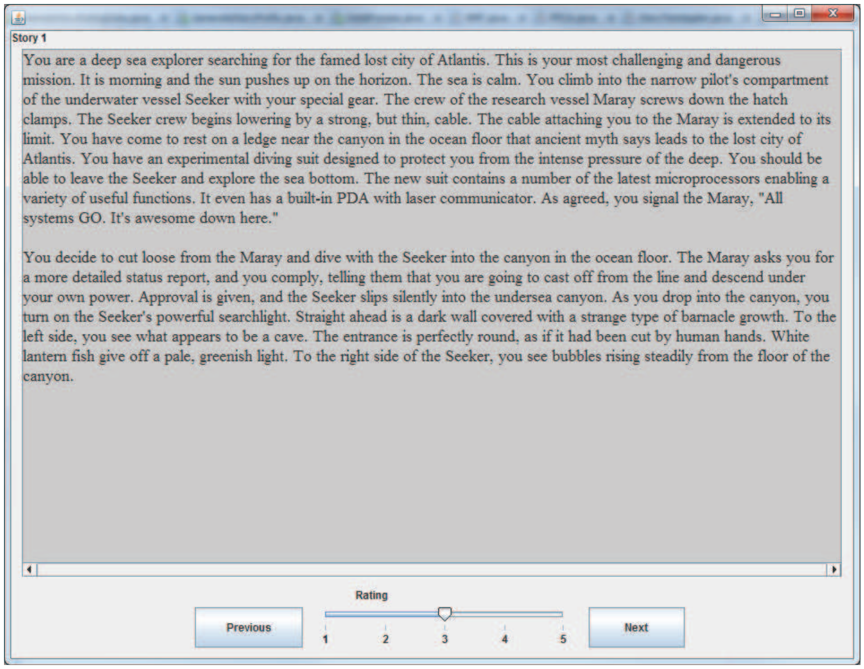

Figure 18: A screenshot of the interactive narrative system. A plot point is a paragraph of the story in the screenshot.

variable of player agency to further facilitate validation of the PBCF algorithm. In Chapter 4, I will build a full interactive narrative system that can provide the players the appearance of full agency.

3.3.3 Model Training on Human Players

In this experiment, I examine the ability of the system to learn a player model for prefixes utilizing human-generated prefix ratings. To learn a model, I need a sparse prefix-rating matrix. To generate the matrix, I implemented a version of the narrative system that randomly walks the branching story graph. It starts at a randomly selected root node and then selects randomly from possible successor plot points. For each plot point presented, the system asks for a rating of the story-so-far from the players.

I recruited 31 players (18 male and 13 female) for the experiment. 26 of the players are college graduate students and the other 5 players are research scientists and staff at our University. Five out of the 31 players were familiar with the choose your own adventure series of books prior to participating in the study. Participants who were not familiar with choose your own adventure books were given a sample adventure story to familiarize them. The experiment took about half an hour for each player.

A 326 by 31 prefix-rating matrix R with $\sim 86\%$ ratings missing is collected in this step. The prefix-rating matrix R was randomly split into training part R^t which contains 90% of the ratings, and validation part R^v which contains the remaining 10% of the ratings. The R^t and the R^v are still of the same dimensions as the original R and both contained missing values.

The NMF and pPCA algorithms are trained on the training set R^t with different parameters. The resulting models were then used to predict the ratings in the validation matrix. To evaluate the accuracy of each algorithm, I measured the Root Mean Square Error (RMSE), which is computed as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{|O|} \sum_{i,j \in O} ((R^v)_{ij} - (R^{v'})_{ij})^2} \tag{11}$$

where $R^{v'}$ is the predicted validation matrix, O is the set of entries indices that are not missing in the validation matrix R^v and $|O|$ is the number of entries that are not missing in R^v.

The random splitting process is repeated ten times and the average RMSEs on the validation sets are reported in Table 1. The dim i in the table mean that from NMF the matrix W in Equation 3 has i columns. The RMSEs in the table suggest that there are probably six types of players in the current training set when it comes to story preferences. Although it is not easy to interpret and label the player types, by learning to cluster participants into these player types the system is able to predict prefix ratings to within one point of actual participant ratings on the story-so-far on average.

Table 1: The average RMSE for NMF and pPCA algorithms with different parameters.

Algorithms	RMSE
NMF_dim3	1.2423
NMF_dim4	1.1781
NMF_dim5	1.1371
NMF_dim6	**0.9901**
NMF_dim7	1.1108
NMF_dim8	1.1354
NMF_dim9	1.2464
pPCA	1.2016

3.3.4 Evaluation of Story Recommendation

I hypothesize that the PCBF algorithm can improve overall ratings of story experiences by increasing the likelihood that players see stories that are more aligned with their preferences than by chance alone. Because of constraints between plot points, I know that all possible paths in a branching story graph from root to leaf nodes were intentionally designed, even random walks produce legal stories from the designer's perspective. I therefore believe that a random walk baseline is a strong baseline. My hypothesis builds on the assumption that individual players may have preferences across the possible story experiences allowable by the designer's branching story graph.

22 graduate students (17 male and 5 female) were recruited to evaluate the PBCF algorithm. None of the participants were involved in earlier experiments. I use the best player model from model training phase (i.e., NMF with 6 dimensions). To compare the model performance on new players versus existing players on which the player model was trained, I also invited 11 players from model training phase back to participate in this study.

The story recommendation study consists of four stages. In the first stage the DM presents five randomly selected stories, generated in the same way as the model

Table 2: The average ratings for the random and personalized full-length stories. The accuracies are the percent of pairs in which the average rating of the personalized stories is higher than the average rating of the random stories.

	Random	Personalized	Accuracy	p-value
All	2.9449	3.8899	0.828	< 0.0001
Returning	3.0320	4.0350	0.863	< 0.0315
New	2.8993	3.8138	0.809	< 0.0001

training experiment. This provides some initial ratings from the participant. In the second stage, five personalized stories are generated according to the PBCF algorithm. These personalized stories are also presented to the participant plot point by plot point and the players' story-so-far ratings are collected after every plot point. As in Sharma et al. [61], the DM generates another five personalized stories in the third stage, followed by five random stories in the last stage in order to eliminate any bias introduced by the order in which the stories are presented to the participants. I do not want the players to feel that the stories in the system become better as they read more stories. In total, every participant is required to read 20 stories (10 total random stories and 10 total personalized stories).

Table 3 shows the results for the new players and returning players. The first line exhibits the statistical results on all the 33 testing player. The second and the third lines give the results of the 11 returning players and the 22 new players, respectively. The first column "Random" and the second column "Personalized" show the average ratings of all the random and all the personalized stories respectively. For every player in the story generation phase, I also compare the pair of average story ratings from the first step and the second step, and the pair of average story ratings from the third and the fourth step. The "Accuracy" column shows the percent of pairs in which the average rating of the personalized stories is larger than the average rating of the random stories, indicating the DM correctly chooses preferred stories. The last column shows the significance of the difference between random and personalized averages using the Wilcoxon signed-rank test.

One reason I would like to collect players' ratings during story recommendation phase is to create a more accurate player model as described in Section 3.2.3. Another advantage is that both the random stories and personalized stories will be presented in the same format so that the players' preference judgement will not be affected by the system interface. In fact, the players were not told which stories were random and which stories were personalized before the experiment and most of them did not notice the difference afterwards. The last reason is that I need to compare their ratings between random stories and personalized stories so that I can evaluate the PBCF algorithm.

3.3.5 Experiment on Human Players without Considering History

Throughout the dissertation, I believe in the theory that history matters in stories. In order to select stories based on player's preference, the drama manager is required to make prediction and recommendation in the context of previously experienced plot points. A story is a recounting of a sequence of plot points. One assumption I have for the theory is that any single plot point taken out of the sequence does not necessarily provide reasonable information for the entire story. Another assumption behind this is that the DM built with player's preference over individual plot points is difficult, if not impossible to catch players' preference on prefixes or full-length stories. The first assumption coincides with our daily experience. To validate the second assumption, I perform another group of human study.

The comparison experiment is designed in which the DM selects the successive plot points based on the players' ratings over plot points. This study was also composed of model training phase and story recommendation phase. In the model learning phase, 19 players participated in the study. Each player read 30 independent plot points without story context. These plot points are randomly selected from the branching story graph and presented to the players one by one. I collected a rating from the players after every plot point as before. Unlike in the prefix based CF human study, the players were told to rate single plot points alone in this experiment. In order to prevent the players from memorizing previous plot points and creating their own story history, I hid previous plot points when presenting the new plot point. And consecutive two plot points were always chosen from different CYOA books. Instead of the prefix-rating matrix, I collected a plot point rating matrix in the model learning phase. Then the player model was built by training the NMF algorithm (with dimension six) on the plot point rating matrix.

In the story recommendation phase, I invited another 14 players for the study. The game interface is similar to the one in story generation phase of the prefix based CF human study. Every player read 20 stories in exactly the same sequence as in previous experiments(5 random, 5 personalized, 5 personalized and 5 random). And after each plot point, the players were required to leave a preference rating for each new plot point. Based on the NMF model learned from the plot point rating matrix, the DM chose the next plot point that can follow the current plot point to guide the players through the branching story graph. Apparently the DM did not take history into consideration when recommending stories.

Notice that the players' ratings are still dependent on the story-so-far since it is impossible for them to forget previous plot points. As a result, in this experiment, the drama manager tries to predict players' preference on story prefixes using the player model built based on players' preference for plot points. Table 3.3.5 shows the results of the comparison experiment.

Table 3: The experiment results of the comparison experiment without considering history.

	Random	Personalized	Accuracy	p-value
All	2.4214	2.5500	0.464	0.4379

3.3.6 Evaluation of Using Robin's Laws as Prior Knowledge

In order to evaluate whether using the Robin's Laws can improve the story preference models I built, I performed another human study which uses the Robin's Laws as prior knowledge. The Robin's Laws player types assume that there are five types of players for games: Fighters (who prefer combat), Power Gamers (who prefer gaining items and riches), Tacticians (who prefer thinking creatively), Storytellers (who prefer complex plots) and Method Actors (who prefer to take dramatic actions) [67, 41]. In order to test the Robin's Laws on the players data, I labeled all the story prefixes in the story library. Each prefix was labeled with a five dimensional vector according to Robin's Laws player types. The entries of the vector, ranging from 0 to 1, express the average belief about how the story prefix matches players of the corresponding Robin's Laws player type. For example, a story prefix with the vector $[0.1, 0, 0, 0.9, 0]$ means most Storytellers will like it very much and there is a slight probability that Fighters will also like it. The labeling of prefixes was performed by three college students, including me. To mitigate bias, the final label for each prefix is the average produced by each of the human labelers. These entries in the vectors are scaled to 1 to 5 and treated as ratings. I get a 326 by 5 rating matrix P, each column of which represents ratings for all the prefixes from a Robin's Laws player type.

As mentioned in Section 3.2, known prior knowledge can be included into the model learning process. In the experiment, I set the first five columns of matrix W in Equation 3 to be P and keep the five columns constant during the EM learning process. I train the model on the rating data from 31 human players in the modeling

Table 4: The RMSE for NMF algorithms with Robin's Laws prior on human player data.

Algorithms	RMSE
NMF_dim5	2.025
NMF_dim6	1.891
NMF_dim7	1.792
NMF_dim8	1.917
NMF_dim9	1.988

learning phase of the first human experiment in Section 3.3.3. The RMSE results for NMF of different dimensions are shown in Table 4. Notice that W is constant for NMF_dim5 algorithm in the table. So I only need to learn H in the model learning phase.

3.3.7 Experiment with Simulated Players

In addition to studies with human participants, I also conducted experiments on simulated computer players in order to get a more complete picture of PBCF algorithm performance. Simulated players are more consistent over time, allowing me to make observations about my algorithms on a controllable data set. With simulated players I can generate larger data sets to examine algorithm learning rates and experiment with different algorithm designs without requiring hundreds of human participants. Note that it is not required that the simulated players play in the same way or have the similar preference as human players. Instead, as long as the simulated players are consistent on their behaviors, they can be used to test the capability of my system to capture players' preference and build player models.

The simulated players are built based on the Robin's Laws player schemes described earlier and used in related works. Every simulated player is created with a five-dimensional characteristic vector. Each entry of the vector (ranging from 0 to 1) specifies the corresponding characteristic of the simulated player. For example, the vector $[0, 0.7, 0, 0, 0.9]$ means the simulated player is a combination of Power Gamer and Method Actor and tends to enjoy Method Actor a little more. Story prefixes are labeled according to beliefs about how they match Robin's Laws player schemes. Three graduate students (including me) labeled each prefix independently. The final prefix labels were computed by averaging the labels from the three students. Note that the prefix labels are not required to be accurate descriptions of the story prefix content since I do not aim to imitate human preference.

Furthermore, I assume that a simulated player will prefer a story prefix that most closely matches the player's type. For example, a simulated player with characteristic vector $\mathbf{p} = [0.8, 0, 0, 0, 0]$ will prefer for a story prefix i with label $\mathbf{s}_i = [1, 0, 0, 0, 0]$ over a story prefix j with label $\mathbf{s}_j = [0, 1, 0, 0, 0]$. Consequently, I assume that the rating r of a simulated player $\mathbf{p}$ for a prefix $\mathbf{s}$ is proportional to cosine distance between the vector $\mathbf{p}$ and $\mathbf{s}$: $r \sim \frac{\mathbf{p}^T\mathbf{s}}{|\mathbf{p}||\mathbf{s}|}$. The ratings are computed by scaling the cosine distances to the range between 1 and 5. In addition, I add random noise with standard Gaussian distribution (mean 0 and variance 1) to all the ratings in order to simulate the variability in the human player behaviors that it could be inaccurate to quantitate preference into digital labels.

In the model learning phase, 120 simulated players were created with characteristic vectors randomly chosen from {[1, 0, 0, 0, 0], [0, 1, 0, 0, 0], [0, 0, 1, 0, 0], [0, 0, 0, 1, 0], [0, 0, 0, 0, 1]}. Each simulated player then "read" 10 random stories and generated a preference rating after every plot point, similar to what I asked of human participants. A 326 by 120 prefix-rating matrix was generated in this way and used to train the player model.

In order to test the generalization ability of the PBCF algorithm, a new group of 1000 simulated players was created in the story generation phase. Each simulated player was given a random characteristic vector, of which the five entries were floating point values ranging from 0 to 1. The story generation phase follows the same four steps as the human story generation experiments. For the purpose of comparison to other algorithms, the DM generated personalized stories using the following algorithms:

- *BaselineP*: using pPCA to learn the player models based only on simulated players' ratings for full-length stories instead of prefixes, then directly recommend the full-length stories instead of choosing branches through recommending prefixes. The BaselineP algorithm behaves similar to a traditional movie recommendation system where full-length movies are recommended based on others' ratings on the full-length movies.
- *BaselineN*: The same as *BaselineP* except using NMF as the CF algorithm.
- *Vector*: Each player is simulated as a vector which initially is $[0, 0, 0, 0, 0]$. For every plot point encountered, the DM updates the characteristic vector based on the features of the current story prefix including the new plot point. Then the DM generates successive plot points by recommending the following prefix based on the updated player vector, or chooses randomly when there is no clear preference. The vector based player modeling algorithm is built to simulate the model learning technique used by Thue et al. [67][2].
- *pPCA*: The prefix based CF algorithm using pPCA; same as with the human players.
- *NMFwoP*: The prefix based CF algorithm using NMF without prior knowledge; same as with the human participant story generation experiment in Section 3.3.4.
- *NMFwP*: The prefix based algorithm using NMF with Robin's Laws player schemes as prior knowledge. In the case of simulated players, I can compute

[2] A difference between Thue's approach and the *Vector* approach is that Thue et al. bootstrap the player model with pre-game players' selections, while I use a uniform non-informative vector as the initial player model [67].

Table 5: The experiment results for the simulated players using several variations of the DM algorithm.

Algorithm	Random	Personalized	Accuracy
BaselineP	2.2190	2.5305	0.668
BaselineN	2.1752	2.4582	0.643
Vector	2.2010	2.8335	0.617
pPCA	2.2350	2.9607	0.798
NMFwoP	2.2362	3.3950	0.894
NMFwP	2.2013	4.0027	0.949

the accurate rating vector $\boldsymbol{w}^j$ for each known player type j, where $j = 1, ..., 5$ correspond to the five player types in the model learning phase. These vectors $\boldsymbol{w}^j$ are included in the matrix W in Equation 3 as fixed columns during the model learning process. This condition represents the near ideal circumstance where the designer has strong genre knowledge about how players respond to stories and can author plot point sequences accordingly.

The experiment results for these algorithms on the 1000 simulated players are shown in Table 5. The results are all statistically significant at p-values approaching zero (using one-tailed t-tests on random and personalized averages) due to the large number of simulated testing players.

I explored the learning rate of different player modeling algorithms. In previous experiments, each player read 5 stories (random or generated) in each of the four steps of the story generation phase. I alter this number and compare the story generation accuracy for different algorithms. Figure 19 shows the average accuracies of 1000 simulated players as the number of stories read in every step changes. As shown in the figure, the NMF algorithms can achieve accuracies higher than 70% even when a new simulated player reads only one story.

The influence of the training set size on the player model learning process was also tested. Figure 20 shows the average RMSEs of the three prefix based algorithms

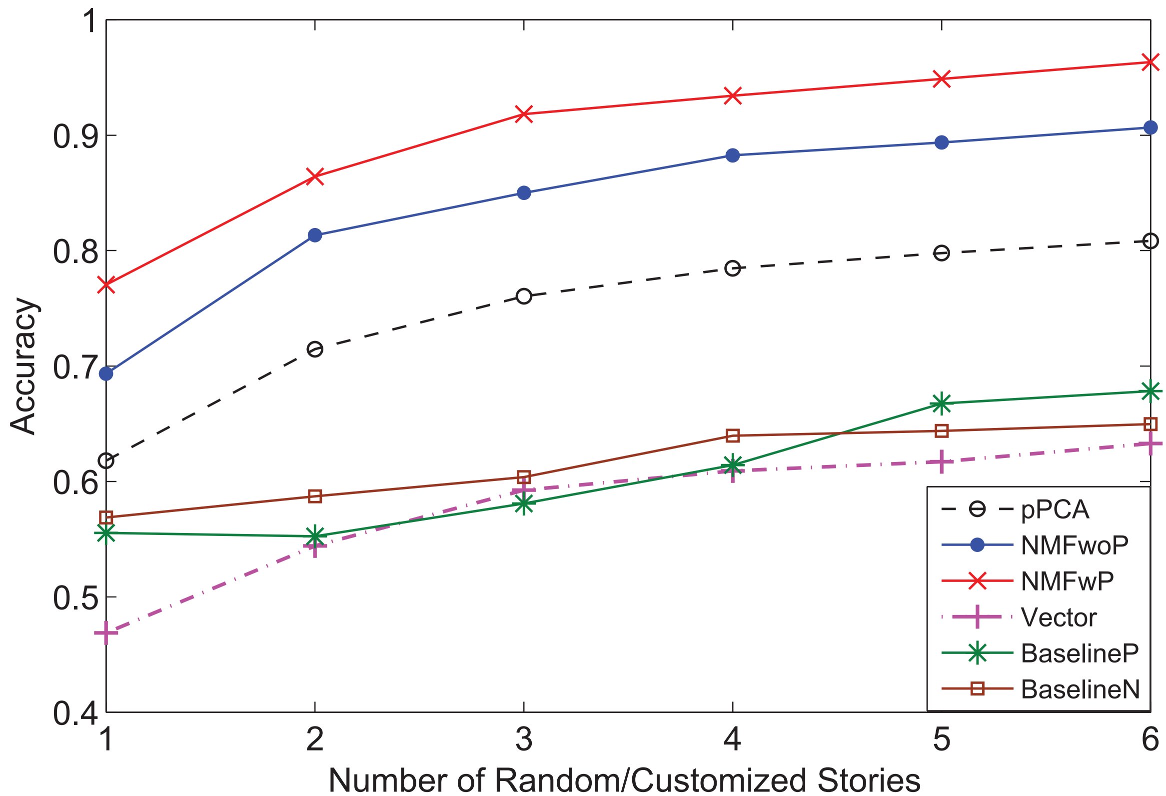

Figure 19: The accuracies of the six algorithms as the number of stories read in every step changes.

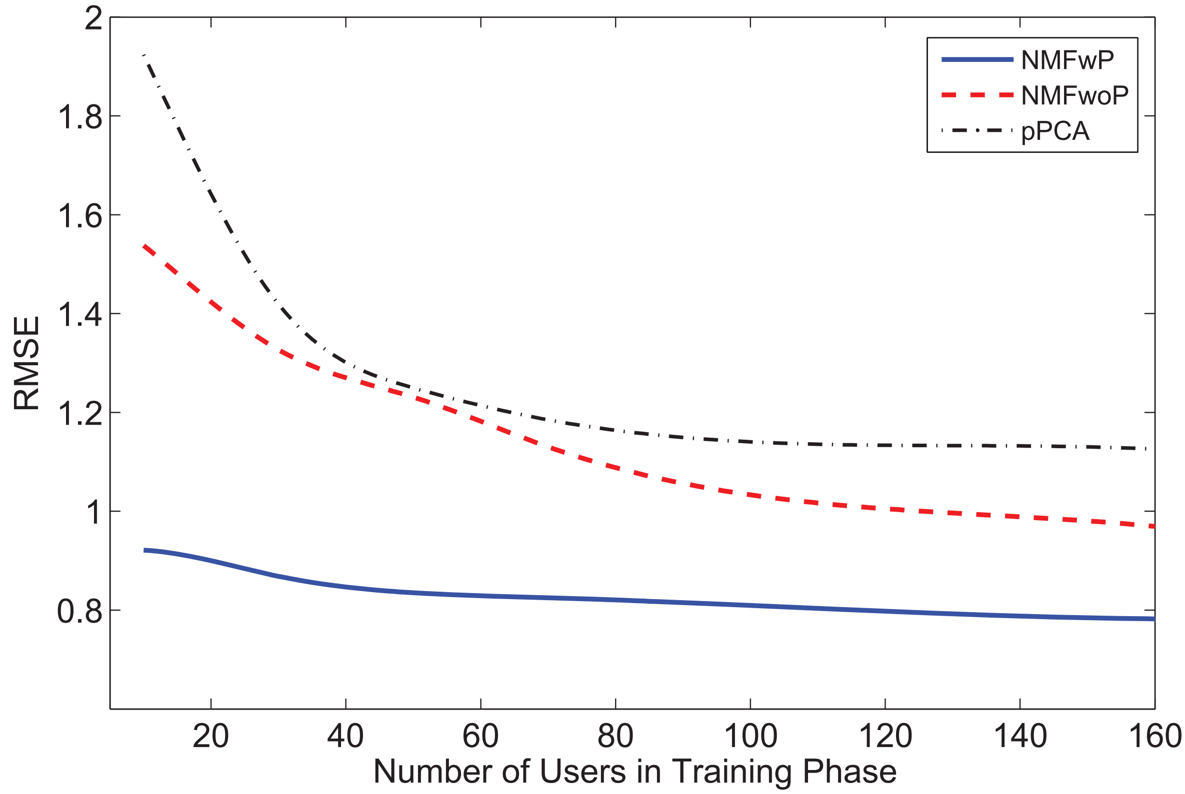

Figure 20: The average RMSEs of the three prefix based algorithms with different number of simulated players for training.

with different number of simulated players for training. Each RMSE value in the figure is an average computed from 10 random splits of the training data. As seen from the figure, the training RMSEs decrease as the training set size grows. Due to the Gaussian noise in the rating data, the RMSE values for the NMFwP algorithm become stable after the number of training players goes above 100 even if it has the perfect prior knowledge of the player models.

3.4 Discussion and Conclusions

The experiment on the human players using the prefix based collaborative filtering algorithm achieves high story recommendation accuracies on the current choose your own adventure data set. In the story recommendation study, the new players rate DM-generated stories higher than random stories for over 80 percent of times. I can achieve this rate after the new players have only read and rated 5 sample stories. The accuracy of about 86% is achieved when the testing players' data are already part of the trained model. For all the participants including the returning players and the new players, The average ratings for the personalized stories are higher than the random stories. And the p-values are approaching zero for the results. The results show that the prefix based collaborative filtering algorithm can capture the players' preference and recommend new stories with high accuracy and validate my hypothesis in Section 3.3.4.

During the experiment on the human players without history in Section 3.3.5, the accuracy is around 56%. The plot point based collaborative filtering algorithm performs similar to a random selection algorithm. And thee average rating of generated stories is approaching the average rating of the random stories. The results demonstrate the assumption that the player model built on plot point preference cannot be used directly to predict players' preference for prefixes or full length stories. Thus the human study proves the importance of history in story generation, the theory based on which I build the prefix based collaborative filtering algorithm.

Using Robin's Laws as prior knowledge for the PBCF algorithm in Section 3.3.6 does not improve the learning accuracy. On the contrary, the learned player model is much worse than the model from the PBCF algorithm without the prior knowledge in terms of RMSEs. The results shows that the real human's ratings are so complex that simple combinations of a few pre-defined player types is incapable of capturing the "patterns" from them. The real "patterns" in the ratings might be so far away from the Robin's Laws model that introducing the prior knowledge could reduce the accuracy.

In the experiments with simulated players, the NMF algorithm usually performs better than the pPCA algorithm. One reason for it might be the linear model assumption for the simulated players. The linear characteristic model for simulated players coincides with the basic assumption of the NMF algorithm, which also assumes that players (columns of the matrix R in Equation 3) are linear combinations of a set of bases (columns of the matrix W in Equation 3). Although NMF is a natural fit for the simulated players, it is also superior to pPCA for human data in terms of RMSEs in my experiments.

Figure 19 illustrates that the prefix based algorithms are capable of extracting player preference much faster than traditional CF algorithms, which create the player model directly on full-length stories (baseline algorithms *BaselineP* and *BaselineN*). The results demonstrate that the players' preference for story prefixes does correlated with players' preference for full-length stories. Compare to the experiments on human players without history in Section 3.3.5, I can conclude that the player model built based on prefix ratings are better at describing the players' preference over full-length stories than the plot point based player model. Another reason the prefix based algorithms work better than traditional CF is that the prefix based algorithms can obtain more preference information (the ratings on all the prefixes) from the players in both model learning and story generation phases. Figure 19 also shows that the Vector approach learns the player model much slower than my algorithms and is thus less accurate on average. This is because the Vector approach cannot acquire any information from the training data.

The total number of prefixes in the story library could be exponential in the number of plot points. But in practice I can effectively add constraints between plot points to limit the size of the prefix database. Notice that in my system, the number of total prefixes grows linearly with the number of total stories given a limit of maximum number of plot points in each story because of the constraints imposed by the branching story graph representation. Although there are only 154 full-length stories and 326 prefixes in the story library, the well-known scalability of collaborative filtering algorithms suggests that the PBCF algorithm can be extended to handle larger scale problems as long as I have enough rating data. In traditional recommendation systems, CF algorithms can easily process products-user matrix with dimension of hundreds of thousands and achieve high recommendation accuracies [64].

In this chapter, I develop a prefix based collaborative filtering algorithm to model players' preference over story prefixes. The PBCF algorithm enables the drama manage to build a flexible player preference model without rigid assumption on pre-defined player types. Compare to other player modeling algorithms with rigid predefined player types, the PBCF algorithm can learn player types dynamically from players' feedback and better capture players' story preference. A simple interactive game is created based on CYOA series to evaluate the PBCF algorithm. The drama manager using PBCF algorithm is capable of choosing successive plot points on behalf of the players and increasing the players' story preference ratings. But in the current system, the players are not allowed to make option selection by themselves. In the next chapter, I will build a fully interactive narrative system that restores player agency.

CHAPTER IV

PERSONALIZED DRAMA MANAGER

In all the human studies until now, the drama manager selects successive story plot points for the players. The drama manager utilizing the PBCF based player modeling algorithm can optimize an individual player's narrative experience by directly selecting the branches through a branching story graph that are predicted to earn the highest ratings. However, this comes at the expense of *player agency*—the ability for the player to determine his or her own narrative future. In this chapter, I will restore player agency to allow the players to make choices by themselves.

Unfortunately, the players do not know what will happen in future plot points when they make choices. They usually select options based on a variety of local clues, such as the words in the options, contents in previous plot points, etc. Given full agency, the players' myopic selection could be in contradiction with their own preference over future story experience, while the DM's selections based on the PBCF algorithm can recommend better stories with high probability, as shown in previous chapter.

In this chapter, I describe a personalized drama manager to address the contradiction. The personalized drama manager has two tasks. The first task is to select target trajectories that maximize the players' expected preference ratings. The prefix based collaborative filtering algorithm can be used to choose the target trajectories. The second task for the personalized drama manager it to increase the probability that the players make choices that coincide with the target trajectories selected by the drama manager. I develop a personalized guidance algorithm to achieve it.

The personalized drama manager works in a *multi-option branching story graph*, in which multiple options are allowed to point to the same plot point. Given the multi-option branching story graph, the personalized guidance algorithm uses collaborative filtering to predict which option a player is likely to choose and only presents a subset of options such that the player is more likely to follow the target branch that the drama manager selects.

This chapter is organized as follows: firstly I will introduce the multi-option branching story graph. Then I will describe the player modeling algorithm to model the players' preference over the options, followed by the personalized guidance algorithm. After that, I will describe how the personalized drama manager selects target successive plot points, followed by a summarization of the personalized drama manager algorithm. At the end, I will describe the human studies I performed to evaluate the personalized drama manager.

4.1 Multi-option Branching Story Graph

The personalized drama manager works in a multi-option branching story graph. Figure 21 shows a simple multi-option branching story graph which is transformed from Figure 3. In Figure 21, each non-leaf plot point has two or three options pointing to each successive plot points. Ideally, there are multiple options between all plot points and their immediate successors in the multi-option branching story graph.

With the multi-option branching story graph, the personalized drama manager can pick a subset of the options to present to the player such that at least one option leads to each child (ensuring true player agency) and also increase the likelihood that the player will pick the option that transitions to the desired target plot point. For example, suppose a player is at plot point 5 in Figure 21 and the personalized drama manager predicts that the player's optimal narrative trajectory is through plot point 8. Suppose the drama manager further predicts that the player's preferences over all the available options to be $g^1 > h^1 > g^2 > h^2 > g^3 > h^3$, such that the player is

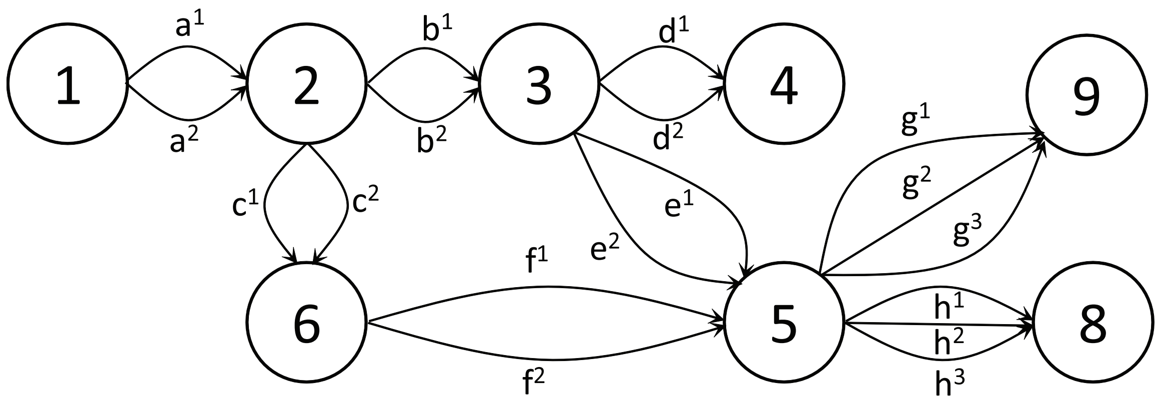

Figure 21: A branching story graph with multiple options pointing to the same plot point.

predicted to transition to plot point 9 instead. To intervene, the personalized drama manager will present options h^1 and g^3 to the player, while suppressing the other options.

This simple extension to the conventional branching story graph gives the drama manager the ability to subtract options from players' considerations without completely pruning a branch of the graph. This preserves the authorial intent behind the structure of the graph and also ensures that all trajectories through the graph are available to the player at all times.

I believe that options should be authored to appeal to different motivations that they players might have, tapping into individual differences. In this research, I utilized the following motivational theories, drawn from Petty [43] and Cialdini [9], to author the additional options:

- *Expert Source*: a desire to follow experts' opinions.
- *Scarcity*: a desire for something that will soon become unavailable.
- *Consistency*: a desire to appear consistent with what we have already done or said.
- *Social Proof*: a desire to imitate others in similar situations.

- *Reasoning*: a desire to follow arguments that sound rational.
- *Number of arguments*: a desire to follow statement that contains repetitive arguments expressed in different ways without new information.
- *Motivation–Friendship*: a desire for friendship.
- *Motivation–Safety*: a desire for being safe.
- *Motivation–Money*: a desire for being rich.
- *Motivation–Fame*: a desire for being famous.

Authoring of options based on the above motivational theories is not strictly necessary, but I hypothesize that utilization of motivational categories will improve the personalized drama manager's ability to learn players' preferences for options. In Section 4.6 and Appendix A, I will describe more details about the multiple option authoring process.

4.2 Player Option Preference Modeling

I assume that players have different preferences over the difference version of options in the multi-option branching story graph. For a particular player, if I know his/her preference for all the options in the multi-option branching story graph, it will be straightforward for the drama manager to select a subset of options to show. In this section, I will build an option preference model to predict which options the player will prefer at any given plot point.

Collaborative filtering algorithms are used to build the option preference model. Applying collaborative filtering algorithms to option preference, I have players rate the options presented after each plot point in the training phase. Then I construct an option-rating matrix similar to the prefix-rating matrix in Figure 13. Figure 22 illustrates a sample option-rating matrix.

Option	Player 1	Player 2	Player 3	...
a^1	*	*	2	...
a^2	1	*	2	...
b^1	4	3	*	...
b^2	*	5	1	...
...	...	...	...	...

Figure 22: An illustration of the option-rating matrix. The stars represent those missing ratings.

An n by m option-rating matrix contains the ratings for n options from m players. Each column of the option-rating matrix contains one player's preference ratings for all the options while each row contains ratings for one option from all the players. The matrix will be sparse, containing a large number of missing ratings since I do not expect each player to read all the options in the extended branching story graph.

I investigated a variety of common collaborative filtering training algorithms on the option-rating matrix, including: NMF, pPCA, K-Nearest Neighbor[1], and K-means algorithms[2]. The NMF and pPCA algorithm are used in a similar way as in Section 3.2.1.

The K-Nearest Neighbor algorithm predicts the option rating vector r of a new player through using option ratings of K nearest training players. To be more specific, r is computed as follows:

$$r = \frac{1}{K} \sum_{i \in O_K} r_i \tag{12}$$

where O_K is the set of K nearest players to the current player. The distance between

[1] http://en.wikipedia.org/wiki/K-nearest˙neighbors˙algorithm
[2] http://en.wikipedia.org/wiki/K-means˙clustering

the new player and the training players can be computed using l_2 distance between the initial option ratings of the new player and the option ratings of the training players.

The K-means algorithm clusters the training players into k clusters through minimizing:

$$\sum_{i=1}^{k}\sum_{r\in S_i}||r-r_i||^2 \tag{13}$$

where S_i is the i^{th} cluster of the option rating vectors and r_i is the center of the option rating vectors of cluster S_i. Given a new player with an initial option rating vector, the drama manager firstly clusters the new player into one or several (if soft clustering is allowed) of the k clusters. Then the cluster center can be used as the prediction of the option ratings for the new player.

For the NMF, pPCA and K-means algorithms, the learned player model retains the extracted rating patterns for players of different option preference types and will be used to predict future players' preference ratings over the options. Once training is complete, the player option preference model can be used to predict players' ratings for options that players have never encountered. This includes the possibility of predicting a player's preferences for options on a graph that he or she has never played through if I have data for the player from another graph.

4.3 Personalized Guidance Algorithm

The personalized drama manager attempts to influence players' trajectories in the multi-option branching story graph such that the players are more likely to select options that leads to the selected target plot points.

In the training step, the personalized drama manager collects option ratings to populate the option-user matrix, which is used to train the option preference model. In the testing step, the personalized drama manager uses the personalized guidance algorithm to guide new players in the story space. For each new player, the personalized DM must collect a few initial option ratings r_o to bootstrap the player model. These ratings can be collected on a graph especially for training on new players or can come from repeated interactions with the system. Once a player is in the option-rating matrix, the personalized DM uses the following personalized guidance algorithm at each plot point:

1. Determine which child plot point the player should experience next.
2. Predict the player's preference for all options using the NMF, pPCA, K-Nearest Neighbor, or K-means algorithm.
3. Display the highest rated option that points to target successor plot point and the lowest rated option for each other successor plot point.
4. Collect player's ratings for the displayed options. Include the ratings into r_o.
5. Player chooses an option.
6. Display the corresponding child plot point according to the player's selection and go to step 1.

It is not strictly necessary to collect ratings as in step 4. I do it in the system for the purpose of collecting as much data as possible to build more accurate player models. With every new rating, the personalized DM will get better understanding of the current player's preference over the options.

4.4 Target Full-length Story Selection

In this section, I describe two algorithms to choose target full-length stories to increase the players' story preference ratings: highest rating algorithm and highest expected rating algorithm.

4.4.1 Highest Rating Algorithm

A simple way to choose the target stories is to select the successive plot points that can lead to the highest rated full-length stories based on PBCF prediction at each branching point in the multi-option branching story graph. But in practice, choosing target based only on PBCF prediction might not be the best solution. It is possible that the personalized DM finds no plot point that can lead to a player preferred narrative experience after it fails to guide the player at some branching point in the branching story graph.

For example, suppose that the player is at prefix node A of the prefix tree in Figure 23, where the multiple options are not drawn for simplicity. The PBCF predicts that a player's preferences over the full-length stories (leaf nodes) G, H, I, J, K, and L are $4, 4, 4, 4, 1$, and 5, respectively. Then the personalized DM will attempt to guide the player to the node 12 if it selects target stories based solely on PBCF predicted story ratings. Let's further assume that after the drama manager intervention, the current player still has a much higher probability to choose the option that transitions to node K instead of L at plot point F for a variety of reasons. In this case, it is very likely that the player will be end up at the node K and receive the worst story experience. A better strategy for the drama manager is to select a full-length story from G, H, I, or J as the target when the player is at node A. Thus the personalized DM needs to take into consideration the probability the player reaches each full-length story when selecting target stories.

4.4.2 Highest Expected Rating Algorithm

The highest expected rating algorithm chooses target plot point based not only on the predicted players' preference over the successive story experience, but also on the probability that the players transition to the selected story experience because the players are given the full agency to make their own choices in the interactive narrative

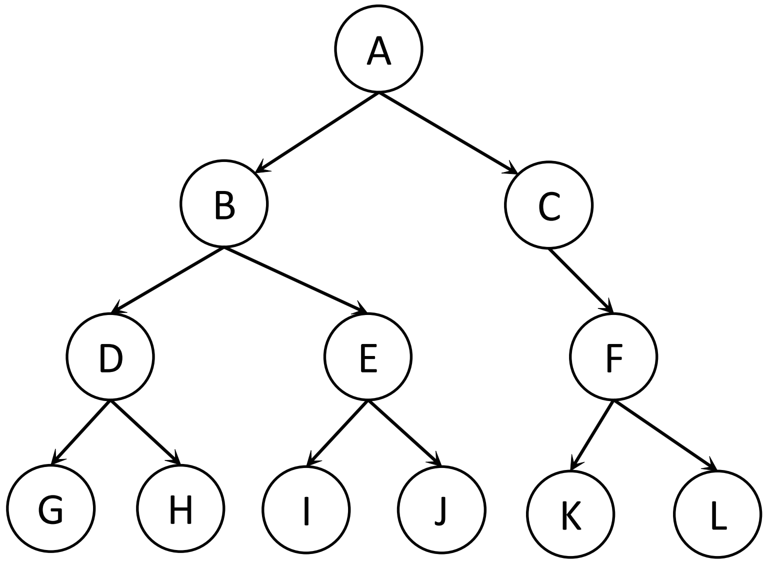

Figure 23: A simple prefix tree. Suppose the player is at prefix node A currently.

system.

To compute the probability that the players reach each full-length story, the personalized drama manager needs to model the players' transition probabilities at each branching point. In the next section, I will describe the branch transition probability modeling algorithm for the personalized DM.

4.4.2.1 Branch Transition Probability Modeling

With the option preference ratings for a particular player, the personalized DM uses probabilistic classification algorithms to predict the player's successive story branch transition probability. Logit regression, Probit regression and probabilistic Support Vector Machine (SVM) are used to train the branch transition probability model.

Logit regression [7] is a probabilistic statistical classification model that can be used to predict the probability that an input data point belongs to each class. The binary Logit regression assumes that the class label y_i for each input data point $\mathbf{X}_i$ follows a Bernoulli distribution with expectation:

$$\mathbb{E}[y_i|\mathbf{X}_i] = \text{Logit}(\boldsymbol{\theta}' \cdot \mathbf{X}_i) \tag{14}$$

where Logit() is the Logit function and $\boldsymbol{\theta}$ contains the parameters to be learned.

The Probit regression model [7] is similar to Logit regression, except that the Logit function is substituted with a Gaussian cumulative distribution function in Equation 14. The probabilistic SVM [44] trains a traditional SVM and an additional sigmoid function that maps the SVM outputs into probabilities.

Applying the probabilistic classification algorithms to the branch transition probability modeling, I define $\mathbf{x}^I_{J,K}$ to be the feature that the player is at a prefix node I with two successive prefix nodes J and K, where the node J is the preferred child selected by the personalized DM. $\mathbf{x}^I_{J,K}$ is a two dimensional vector containing highest preference rating for the options transitioning to the preferred node J and the lowest preference rating for the options transitioning to the node K. To be more specific, $\mathbf{x}^I_{J,K}$ is:

$$(\max_{\alpha \in \mathbb{O}^I_J}\{\mathrm{R}(\alpha)\},\ \min_{\beta \in \mathbb{O}^I_K}\{\mathrm{R}(\beta)\})' \tag{15}$$

where $\mathrm{R}(\cdot)$ is the predicted preference rating for an option, $\mathbb{O}^I_J$ is the set of options that lead to preferred successive prefix node J from node I, and $\mathbb{O}^I_K$ is the set of options that lead to the other successive prefix node K from node I.

The probability $\mathrm{P}^I_{J,K}$ that the player transitions from I to J under the DM intervention is:

$$\mathrm{P}^I_{J,K} = \mathrm{f}(\mathbf{x}^I_{J,K}; \boldsymbol{\theta}) \tag{16}$$

where f could be the Logit, Probit, or probabilistic SVM model, $\boldsymbol{\theta}$ are the parameters to be learned. Notice that $\mathrm{P}^I_{J,K} + \mathrm{P}^I_{K,J} \neq 1$ due to the DM intervention. For a prefix node that has three or more successive nodes, a multinomial Logit regression, multinomial Probit regression or multi-class SVM can be used in a similar way to model the transition probability P [7].

For example, suppose a player is at prefix node A in Figure 23 and the DM selects node C as the desired target for the player. Suppose that the DM has six options (α^1, α^2, and α^3 point to node B and β^1, β^2, and β^3 point to node C. Then the feature value $\mathbf{x}^A_{C,B}$ contains the maximum of the three preference ratings for options β^1, β^2, and β^3, and the minimum of the three preference ratings for options α^1, α^2, and α^3. The probability $\mathrm{P}^A_{C,B}$ will be $\mathrm{f}(\mathbf{x}^A_{C,B}; \boldsymbol{\theta})$.

For a player at node I, I define $\mathbb{P}^I_L$ to be the probability that the player transitions to a leaf node L under the DM intervention. $\mathbb{P}^I_L$ can be computed by multiplying the successive transition probabilities through the path from node I to node L. For example, in the Figure 23, suppose the player is at the root node A. The probability that the player transitions to node L: $\mathbb{P}^A_L = \mathrm{P}^A_{C,B} * \mathrm{P}^F_{L,K}$.

4.4.2.2 Target Full-length Story Selection

For a player at node I of a prefix tree, the personalized DM will select an target full-length story from the subgraph with the root I to maximize the player's expected enjoyment. More precisely, the personalized DM selects a leaf node L^* such that:

$$L^* = \mathrm{argmax}_{L_i \in \mathrm{Leaf}^I} \{\mathrm{R}(L_i) * \mathbb{P}^I_{L_i}\} \tag{17}$$

where Leaf^I is the set of leaf nodes (full-length stories) in the subtree with root I in the current story prefix tree; $\mathrm{R}(L_i)$ is the predicted story rating for L_i using PBCF; $\mathbb{P}^I_{L_i}$ is the predicted probability that the player transitions to L_i from the current node I under the DM intervention as computed in the previous section.

4.5 Personalized Drama Manager Algorithm

The personalized drama manager puts the story preference model (PBCF), the personalized guidance algorithm and the target story selection algorithm to use as follows. For a new player, the personalized drama manager must first collect a few initial ratings for story prefixes and options. These ratings can be collected on a graph especially for training on new players or can come from repeated interactions with the system. The collected ratings are then used to bootstrap the PBCF model and the CF model for option rating prediction. Then at each prefix node I in the

1: **while** I is not a full-length story **do**
2: Predict the ratings for full-length stories L_i that are descendants of I using PBCF
3: Predict the ratings for all the available options in the subtree with I as its root using CF
4: Calculate the probabilities that the player transitions to each L_i under DM intervention: $\mathbb{P}^I_{L_i}$
5: Select an objective full-length story L^* that has the highest expected rating using Equation 17
6: Increase the probability the player transitions to the successive node that leads to L^* by showing a subset of options to the player
7: Collect the player's preference over the story-so-far (the current node I) and the presented options and update the PBCF and CF models
8: The player chooses an option
9: Set I to be the successive prefix node based on the player's selection
10: **end while**

Figure 24: The personalized drama manager algorithm.

prefix tree, the personalized drama manager uses the algorithm in Figure 24 to guide the player.

Notice that it is not strictly necessary to collect story ratings and option ratings as in step 7. I do it in my system for the purpose of collecting as much data as possible to build more accurate player models. With every new rating, the personalized drama manager will get better predictions in step 2 and 3. On the other hand, if the personalized drama manager does not collect new ratings, it will not be necessary to re-predict the ratings for full-length stories and options after every plot point.

4.6 Evaluation of the Personalized DM

To evaluate how the personalized drama manager can guide the players in the story space and whether the personalized drama manager can increase players' preference ratings, I conducted a group of human studies in an interactive narrative system built with choose your own adventure stories.

I hypothesize that the personalized DM algorithm will be able to significantly affect the behavior of players and significantly increase the players' self-reported story preference ratings, as compared to a version of the interactive story with no drama management. In this section, I will describe the story library, online game environment, methodology, and results.

4.6.1 Stories and User Interface

The branching story graphs are created in a similar way as in Chapter 3. Right now I only use two Choose-Your-Own-Adventure books: *The Abominable Snowman* and *The Lost Jewels of Nabooti*, into two branching story graphs. The branching story graph of *The Abominable Snowman* contains 26 leaf nodes and 19 branching points. The branching story graph of *The Lost Jewels of Nabooti* contains 31 leaf nodes and 18 branching points. The two branching story graphs are converted into two prefix trees. In total I have 134 story prefix nodes in the two trees.

I authored two additional options for every branch in the branching story graphs. Each new option was constructed by rewriting the existing option with different motivations as described in Section 4.1. No new information is added into the new options to ensure that the players' option selection behavior is not influenced by the new information. In the final extended branching story graphs, there are thus three different options per successor plot point at every branching point. In total, there are 275 options in the two branching story graphs.

In the experiments, all the stories were presented plot-point by plot-point to the players. After each plot point, the players were asked to rate the story-so-far (for PBCF training) and all the options (for option-preference CF training) on a scale of 1 to 5 before they could select one of the options to continue. A larger rating number indicates a greater preference. Figure 25 shows the online interactive narrative testbed. The figure shows two plot points, a place for players to rate the story-so-far (for PBCF training), and two options with ratings (for option-preference training).

The human study is composed of two phases: model training and testing, which will be described in the following sections.

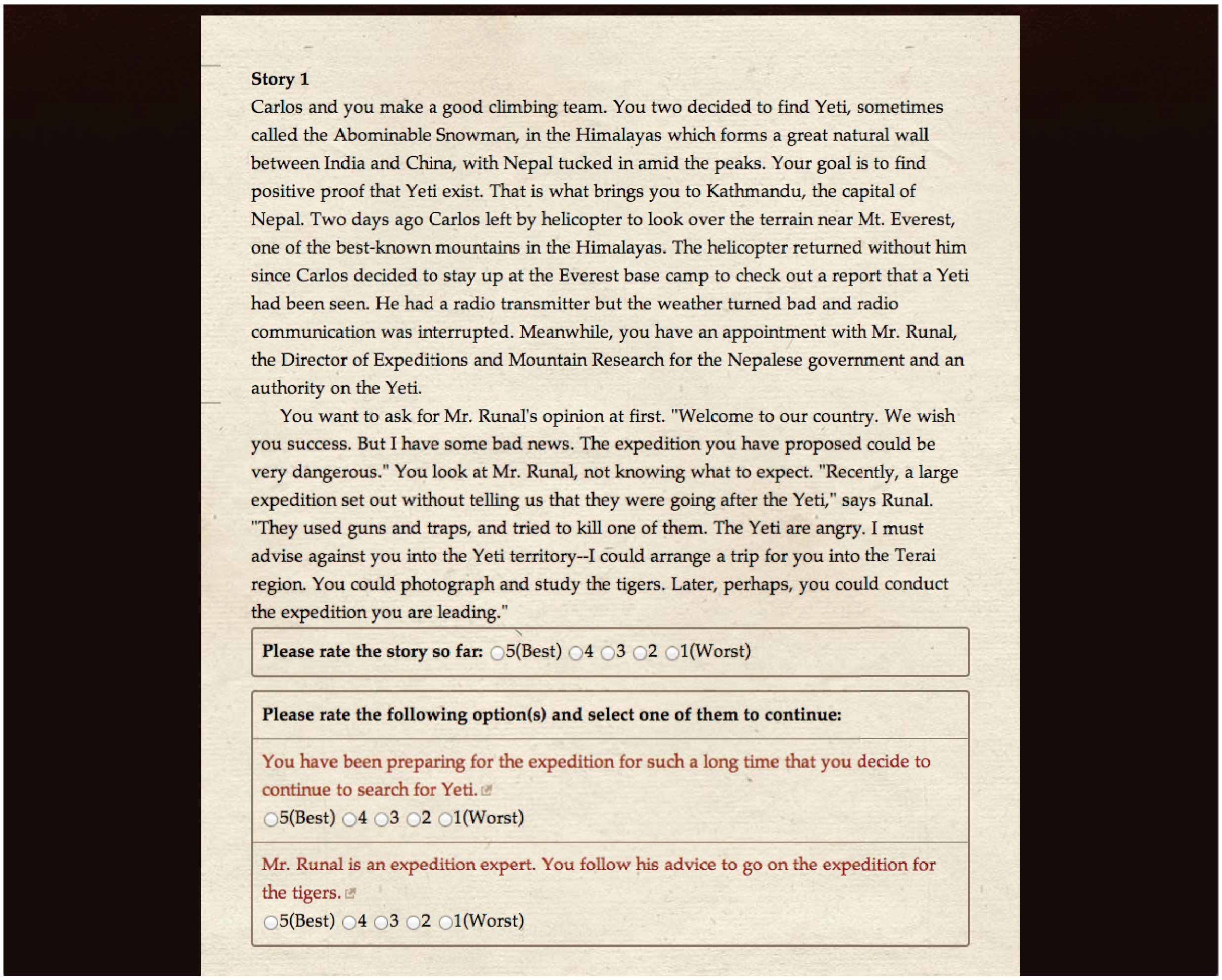

Figure 25: A screenshot of the interactive narrative system.

4.6.2 Training the Personalized Drama Manager

I recruited 80 participants from Amazon's Mechanical Turk. Each player read 4 to 6 full-length stories, each of which was randomly started at the root of one of the two branching story graphs. In total I had 410 valid play-throughs from the 80 players. Each story was presented plot-point by plot-point to the player. At every branching plot point, the DM randomly picked one option for each successor plot point to present to the player and the player was free to make a choice. I collected the players' ratings for all the options and stories they read. The players were asked to explore the graph as much as possible. If the players encountered a plot point they had seen previously, their previous ratings for story-so-far and options were automatically filled out from their previous response. I obtain a 134 by 80 prefix-rating matrix and a 275 by 80 option-rating matrix in the training process.

To train the PBCF model, I randomly select 90% of the ratings in the prefix-rating matrix to train the pPCA and NMF algorithms, which are then used to predict the remaining 10% of ratings in the prefix-rating matrix. The process is repeated 50 times. I compute the root-mean-square-error (RMSE) between the predicted prefix ratings and the real players' prefix ratings on the remaining 10% of the data. The best average RMSE for pPCA algorithm (dimension 46) is 0.576 (a fraction of a rating), and for NMF algorithm (dimension 12) is 0.743 (a fraction of a rating). Thus pPCA is used to model players' story preference in the testing phase. Note that the results are different from the PBCF results in Chapter 3, where NMF achieved better results because I derived a different version of pPCA algorithm using EM algorithm.

To train the option preference model, I randomly select 80% of the training players to learn an option preference CF model. For the remaining 20% of players, the DM builds the initial rating vector from the players' option ratings in one of the branching story graph and predicts option ratings in the other branching story graph. I repeated the process 50 times. The best average RMSE for pPCA algorithm is 0.550 (dimension

Table 6: The training results of the branch transition probability model using three probabilistic classification algorithms.

Algorithm	θ_0	$\boldsymbol{\theta_1}$	Accuracy
Logit	0.086	(1.151, -1.092)	78.89%
Probit	0.099	(0.645, -0.622)	78.19%
PSVM	-	-	79.35%

225), and for NMF algorithm is 0.798 (dimension 9). Thus the pPCA algorithm is also selected for option preference modeling in the testing phase.

I train the branch transition probability model using the predicted option ratings from the learned option preference model and the players' option selection. Table 6 shows the training results for the three probabilistic classification algorithms: Logit regression, Probit regression and probabilistic SVM. The column θ_0 and $\boldsymbol{\theta_1}$ show the parameters from maximum log-likelihood estimation for the Logit and Probit algorithm. Similar to option preference model learning, I randomly select 80% of the training players to learn an option preference CF model. For the remaining 20% of players, the personalized DM firstly builds the initial rating vector using the players' option ratings from one of the branching story graph. Then the DM uses the learned option preference model and the learned branch transition probability model to predict players' branch selection in the other branching story graph. The last column *Accuracy* in Table 6 shows the average prediction accuracies for the three algorithms. The probabilistic SVM algorithm, which uses a radial basis as its kernel function, achieves the best accuracy among the three classification algorithms. However, I select Logit regression for branch transition probability modeling in the testing phase because the linear model is more stable against the noise in the predicted option ratings.

4.6.3 Testing the Personalized Drama Manager

I recruited another group of players from Mechanical Turk to evaluate the personalized DM's ability to increase the players' enjoyment. Each player read 6 full-length stories plot-point by plot-point. For the first five stories, the player explored one of the two branching story graphs. As in the training phase, the DM randomly picked one option for each successive branch to present. The story and option ratings collected were used to bootstrap the preference models for the new player. For the sixth story, the player played through the other branching story graph. At each branching point, the personalized DM selected a desired successive plot point and picked a subset of options using one of the three guidance algorithms described below to attempt to increase the player's enjoyment.

In this section, I will first compare the performance of the three target plot point selection algorithms. Then I will evaluate the DM's performance when it picks one option for each successive branch.

4.6.3.1 Personalized Drama Manager Algorithm Comparison

For the purpose of comparison, I implemented the following three target selection algorithms for the personalized DM:

- *HighestRating (HR)*: at each node in the prefix tree, the personalized DM selects an objective full-length story based on predicted ratings of the full-length stories.
- *HighestMeanRating (HMR)*: at each node in the prefix tree, the personalized DM selects a successive node that leads to the full-length stories with the highest mean rating. For example, suppose a player is at node A in Figure 23. The DM will compare the average predicted rating for nodes G, H, I, and J to the average predicted rating for nodes K and L. If the former one is bigger, the DM will select node B as its objective. Otherwise, the DM will select node C as its objective.

- *HighestExpectedRating (HER)*: at each node in the prefix tree, the personalized DM selects an objective full-length story that achieves the highest expected rating. This is my personalized DM algorithm as in Figure 24.

The *HMR* algorithm and the *HR* algorithm are two baselines that the personalized drama manager algorithm (*HER*) will be compared to. I hypothesize that (1) the *HER* algorithm can significantly increase players' preference ratings, compared to the case without drama manager; (2) the *HER* algorithm will result in higher story preference ratings than the *HMR* or the *HR* algorithm.

In the human study, the above three target selection algorithms use the same PBCF story preference model and option preference model for the purpose of comparison. At each branching point, the personalized drama manager used one of the three algorithms to select a desired successive plot point and picked two options for the desired plot point and one option for each other successive plot point.

I recruited 28 players for the guidance algorithm *HR*, 26 players for the guidance algorithm *HMR*, and 47 players for the guidance algorithm *HER*. Table 7 shows the results of the comparison of the three algorithms. The *Algorithm* column refers to the algorithms that the DM is using. The column (*no_DM*) and the column (*DM*) show the average full-length story ratings for stories that are without drama manager guidance (average ratings in the first five trials) and with drama manager guidance (average ratings in the sixth trial). The Wilcoxon signed-rank test is used to compare the ratings for *no_DM* stories and *DM* stories. The p-values are shown in the second to the last column. The last column shows the percent of the time the players chose the options transitioning to the desired plot points selected by the drama manager. As we can see from Table 7, the personalized drama manager algorithm *HMR* and *HER* can significantly increase the players' preference rating for their story experience. The *HER* algorithm has a much higher guidance success rate than the other two algorithms. Note that I recruited more players for the *HER* algorithm than the *HMR*

Table 7: The comparison of the personalized drama manager with three target selection algorithms.

Algorithm	no_DM	DM	p-value	Success rate
HR	3.41	3.50	0.263	65.7%
HMR	3.27	3.62	0.023	64.6%
HER	3.14	3.96	<0.001	81.3%

and the *HR* algorithm in order to perform comparison in Section 4.6.3.2. In fact the *DM* ratings for *HER* was 4.13 ($p<0.001$) after I recruited only 24 players.

I further compared the players' ratings for with DM stories (column *DM* in Table 7) under the three different DM algorithms. The results show that the ratings for the *HER* algorithm are significantly higher than the *HR* algorithm ($p=0.037$). The rating comparisons for *HER* vs. *HMR* and *HMR* vs. *HR* are not significant on a significance level of 0.05 (the p values are 0.126 and 0.452, respectively).

The players' ratings could be unstable in the first few training stories when the players play the interactive narrative for the first time. I perform a sensitivity analysis of the players' story ratings as different numbers of training stories discarded. Table 8 shows the players' average story ratings in the bootstrapping phase as I discard the players' ratings from the first 0 to 3 training stories. The story ratings in second column of Table 8 are the same as the second column in Table 7 because no training story is discarded. The four p-value columns show the comparison of the average ratings of the random stories (with different number of stories discarded) and the average ratings of the personalized stories under different DM guidance algorithms. The sensitivity analysis results in Table 8 shows that the average story ratings stay stable even after I discard the ratings from the first few training stories in the bootstrapping phase.

Table 8: The sensitivity analysis of the players' ratings in the bootstrapping phase as the numbers of discarded training stories change.

	# of discarded stories				p value			
	0	1	2	3	0	1	2	3
HR	3.41	3.35	3.33	3.18	0.263	0.206	0.181	0.071
HMR	3.27	3.23	3.15	3.12	0.023	0.021	0.014	0.012
HER	3.14	3.17	3.17	3.21	<0.001	<0.001	<0.001	<0.001

4.6.3.2 Select One Option Per Branch

In the above human studies, the personalized drama manager picked two options for the desired branch but only one option for all the other successive branches. If the players do not have different preference over different options or the players select the options randomly, they also have higher probability of transitioning to the branches with two options pointing to them. On the other hand, one might guess that the player can infer where the DM wanted him or her to go based on the number of the options if the DM select different number of options for different successive branch. Thus I perform another group of human study to evaluate whether the personalized drama manager would perform differently if it picked equal number of options for each successive branch.

I recruited another 50 players from Mechanic Turk. The study was conducted in the same fashion as in the above testing process. The only difference was that the personalized drama manager picked *one* option for each successive plot point in the sixth trial. The *HER* algorithm was used to guide the player in the sixth trial. The average ratings for full-length stories *no_DM* and *DM* are 3.28 and 3.74, respectively. The *DM* ratings are significantly higher than the *no_DM* ratings (p=0.004). The average guidance success rate is 70.8% for all the 50 players. Thus the personalized drama manager with the *HER* algorithm can also increase the players' preference ratings significantly when the drama manager picks one option for each successive branch.

4.6.4 Player Agency Study

To further evaluate players' experience when interacting with the personalized DM, I performed another group of human study to examine the players' sense of agency in the interactive narrative. To be more specific, I aim to test whether and to what extent the personalized DM algorithm impacts on the players' sense of agency and replayability in the interactive narrative.

In the human study, each player first reads five stories plot-point by plot-point in the same way as in the previous section. Then in the sixth story, I compare the following three cases:

- *no_Choice*: the player has no choice after each plot point. Instead, the personalized DM selects only one option that can lead to the highest expected full-length story at each branching point. This is the baseline case.
- *no_DM*: the player can make his/her own choices. The personalized DM randomly selects one option for each successive branch after each plot point.
- *DM*: the player can make their own choices. The personalized DM selects a subset of options that maximize the probability the player transitioning to the full-length stories that achieve the highest expected preference ratings. This is exactly our personalized DM algorithm.

In terms of impact on the players' experience in the interactive narrative system, I hypothesize that (1) the case *DM* (our personalized DM algorithm) is NOT significantly different from the case *no_DM*; (2) the case *DM* is significantly different from the case *no_Choice*. I hypothesize that the case *DM* is not significantly different from the case *no_DM* because the players are not prevented from make choices to transition to any successive plot points in the branching story graph and they are not aware of the multiple options in the testing branching story graph which they only visit once in the testing phase.

To evaluate the players' experience, in particular their sense of agency, I require all the players to answer 15 yes/no questions about their experience in the sixth story after they finish the game. The questionnaire is shown as follows:

- *Q1:* I felt my choices had impact on later events in the story.
- *Q2:* I felt that I had control over aspects of the story that I wanted control over.
- *Q3:* I felt I could take an active role in the story.
- *Q4:* I felt that I had little influence over the things that happened in the story.
- *Q5:* Sometimes I felt that I didn't have enough control over the direction of the story.
- *Q6:* What happened in the story was my own doing.
- *Q7:* I don't think that chance played an important role in the story.
- *Q8:* The story ending resulted from some unseen forces.
- *Q9:* I felt that the actions I took were meaningful within the context of the story.
- *Q10:* I was able to see the results of my actions.
- *Q11:* I felt that the story would have been different if I had selected different choices.
- *Q12:* I felt that several choices would lead to the same thing happening in the story.
- *Q13:* I felt that the story system tried to push me to some particular story ending.
- *Q14:* Sometimes I felt that the system wanted me to pick some particular choices.

- *Q15:* I would like to play the game again.

Question 1 to question 10 focus on the players' perceived sense of agency (Q1-Q8 are mostly based on locus of control questions [53]. Question 9 and question 10 are specially created based on Murray's definition of agency [38]). Question 11 and question 12 are used to evaluate whether the multi-option branching story graph has impact on the players' experience. Question 13 and question 14 are used to evaluate whether the guidance of our personalized DM has explicitly impact on the players' experience. Question 15 evaluates the replayability for our interactive narrative. Q11 to Q15 are custom-built.

I recruited 22 players for the case *no_Choice*, 23 players for the case *no_DM* and 23 players for the case *DM*. Table 9 shows the results of the player experience study. The first three columns (*no_Choice*, *no_DM* and *DM*) show the percentage of players who answered yes to the corresponding question under the three cases. The column *no_Choice vs. DM* shows the p values from comparing the case *no_Choice* with the case *DM*. The last column *no_DM vs. DM* shows the p values from comparing the case *no_DM* with the case *DM*. As shown in Table 9, there is no statistically significant difference between the case *no_DM* with the case *DM* for all the 15 questions. For the comparison between the case *no_Choice* and the case *DM*, the players' response are significantly different for all the questions except question 8, question 12 and question 15. I compute the Pearson product-moment correlation coefficient[3] for the 15 questions to evaluate the linear correlation between *DM* and *no_Choice*, and between *DM* and *no_DM*. The Pearson product-moment correlation coefficient is -0.675 for *DM* vs. *no_Choice*, and 0.957 for *DM* vs. *no_DM*. It shows that the players responses in the case of *DM* and *no_DM* have very high positive correlation with each other.

[3]http://en.wikipedia.org/wiki/Pearson˙product-moment˙correlation˙coefficient

Table 9: The results for the player experience study. The significant comparisons are mark with * (p value < 0.05).

	no_Choice	no_DM	DM	no_Choice vs. DM	no_DM vs. DM
Q1	36.36%	95.65%	100.00%	0.000*	0.364
Q2	22.73%	82.61%	95.24%	0.000*	0.227
Q3	40.91%	86.96%	90.48%	0.000*	0.409
Q4	72.73%	30.43%	23.81%	0.001*	0.727
Q5	86.36%	34.78%	47.62%	0.013*	0.864
Q6	13.64%	73.91%	76.19%	0.000*	0.136
Q7	77.27%	34.78%	23.81%	0.000*	0.773
Q8	54.55%	60.87%	42.86%	0.458	0.545
Q9	18.18%	91.30%	95.24%	0.000*	0.182
Q10	45.45%	91.30%	100.00%	0.000*	0.455
Q11	59.09%	100.00%	95.24%	0.003*	0.591
Q12	50.00%	43.48%	28.57%	0.181	0.500
Q13	81.82%	30.43%	14.29%	0.000*	0.818
Q14	81.82%	43.48%	28.57%	0.001*	0.818
Q15	31.82%	56.52%	38.10%	0.608	0.318

4.7 Discussion and Conclusions

The learned parameters $\boldsymbol{\theta_1}$ and θ_0 for the Logit model ((1.151, -1.092) and 0.086) and for the Probit model ((0.645, -0.622) and 0.099) show that the players have a higher probability to choose the option that is rated higher than the other options presented by the drama manager, which is aligned with my expectation. The Logit model is capable of correctly predicting the players' branch transitions for 78.9% of the time. Although the non-linear probabilistic SVM classifier achieves higher predicting accuracy on the training data, the generalization error will probably not be reduced due to the prediction error in the option ratings. Thus in the testing process, I use the linear Logit model which is more robust to the noise in the predicted players' ratings. I believe that the input features are more important factors in predicting players' option selection. In the future, it may improve the drama manager's ability to influence the players' choices if I include some personalized features such as the player's previous transition behaviors into the branch transition probability modeling process.

By incorporating the players' transition probabilities into the drama manager's decision process, the personalized drama manager significantly increases the players' enjoyment in the interactive narrative system. The personalized drama manager using *HER* target selection algorithm significantly increases players' preference ratings. *HER* also beats both *HR* and *HMR* in terms of the players' enjoyment ratings. These human study results verify the two hypotheses in Section 4.6.3.1. The guidance success rate of *HER* is also higher than *HR* and *HMR* because the *HER* algorithm does not select targets that the players have low chance to reach. The DM rating comparison between *HER* and *HMR* is not significant. One possible explanation is that I do not have enough testing players, which is suggested by the fluctuation of the players' average ratings in the case of *no_DM* (column *no_DM* in Table 7). The sensitivity analysis in Table 8 shows that the average story ratings stay stable even after I discard the ratings from the first few training stories in the bootstrapping phase.

The human study results show that the personalized drama manager is capable of predicting an individual player's preference over the stories and options, modeling the probability the player transitioning to successive plot points, selecting an objective story experience that can maximize the player's expected enjoyment, and significantly increasing the players' preference ratings, compared to a no-DM case or DM with different target selection algorithms. Although the personalized DM algorithm is studied in a simple testbed, it represents one of the most important fundamentals of drama management: guiding the players to a better experience in a branching story graph. The personalized DM can be easily extended to other story-based computer games and tutoring systems in which the players can select options or perform actions to change the direction of the story progression.

The player experience study results validate both hypotheses in Section 4.6.4. The results show that there is no significant difference between the case *DM* and the case *no_DM* in terms of the players' response to the 15 questions. Thus I did not detect any significant negative impact on the players' experience for the personalized DM algorithm. Compared to the case *no_Choice*, the personalized DM achieves significantly different results for most of the 15 questions. Thus the personalized DM is capable of guiding the players in the story space without being noticed by the players. In other words, the personalized DM improves the players' experience while preserving the players' perceived sense of agency in the interactive narrative system. The comparison results for question 8 and 12 are not significant, probably because the statement of question 8 is not very clear for some players and question 12 is not applicable to the case *no_Choice*. The answers to the replayability question (Question 15) are not significantly different for both comparisons. The replayability does not have obvious correlation with the hypothesis about the player agency. In fact, the question 15 itself could have different ways of interpretation for different players. Without further study, I do not even know whether a significant difference for question 15 means a better result for the personalized drama manager or not. A significant difference between *DM* and *no_DM* may be result from a better story experience for the case *DM*. But a significant difference might also suggest that the players could detect the difference between the two cases. Further studies are required to determine whether the difference is related to the player agency, the player enjoyment, or other elements in the interactive narrative (e.g. the quality, the size of story space).

Although the personalized DM algorithm is studied in a simple testbed, it represents one of the most important fundamentals of drama management: guiding the players to a better experience in a branching story graph. The personalized DM algorithm can be easily extended to other story-based computer games and tutoring systems in which the players can select options or perform actions to change the direction of the story progression. The personalized drama manager is capable of significantly improve players' experience while preserving the players' sense of agency.

CHAPTER V

FURTHER IMPROVEMENT OF THE PERSONALIZED DRAMA MANAGER

In this chapter, I propose two directions of further improvement for the personalized drama manager: reducing the requirement for the amount of training data for the personalized DM and incorporating author's preference into the decision of the personalized drama manager. The following two sections will describe the two improvements, respectively.

5.1 Reduce the Requirement for the Amount of Training Data

In the current system, the personalized drama manager needs to collect a few initial prefix and option ratings after each plot point to bootstrap the player models for each new player. The feedback collection is intrusive and could negatively impact the players' experience in the interactive storytelling system. Unfortunately, it is difficult to acquire the players' story preference information without explicit questionnaire in an interactive storytelling system.

In this section, I first perform human studies to quantitatively evaluate the impact of feedback collection on players' experience and rating quality. Then I will develop an algorithm based on active learning to reduce the amount of training data for the personalized drama manager.

5.1.1 Impact of the Amount of Data Collection on Player Experience

In this human study, I aim to explore the impact of the number of training stories and the amount of collected ratings on the players' experience. I recruited three groups of players. All the players in the three groups read the stories plot point by plot point as in Section 4.6. Each player in group 1 and group 3 reads six stories, the first five of which are training stories and the last one is testing story. Each player in group 2 reads four stories, the first three of which are training stories and the last one is testing story. For group 1 and group 2, their ratings for all the story-prefixes and options were collected. For group 3, the players were not required to rate any story prefix or option except the full-length stories.

For each participant, the training stories are selected from one branching story graph, e.g. the Yeti branching story graph, while the testing story is selected from the other branching story graph, e.g. the Nabooti diamond branching story graph. Prior to the human study, the players were told that the first 3 or 5 stories they were going to read were used to train the AI system and a good training would give them better experience later. I recruited 101 players from Amazon Mechanical Turk, of which 35 players were assigned into group 1, 45 players were assigned into group 2, and 21 players were assigned into group 3. After the players finished all the four/six stories, I asked them three boredom related questions.

- *Q1:* Do you feel bored after reading the training stories?
- *Q2:* After how many training stories you start to feel bored?
- *Q3:* How many training stories you are willing to do for the next time?

The answer to the first question is on a scale of 1 to 5 with higher value meaning less bored. The second and the third questions are multiple-choice questions. Table 10 shows the average response to the three questions for the first two groups of players. As shown in Table 10, the players in group 2 who read 3 training stories felt less bored than the player in group 1 who read 5 training stories on average. The players in group 1 started to feel bored later than group 2. It might suggest that participants were thinking back and choosing a value roughly 20-30% less than the total number

Table 10: The average response to the three boredom related questions for different number of training stories.

	Average boredom	After # training stories players start to feel bored	# training stories for the next time
Group 1 (5 training)	3.27	3.64	3.44
Group 2 (3 training)	3.74	2.51	2.94
p-value	0.301	< 0.01	0.068

Table 11: The average response to the three boredom related questions for different amount of feedback collected.

	Average boredom	After # training stories players start to feel bored	# training stories for the next time
Group 1 (More Feedback)	3.27	3.64	3.44
Group 3 (Less Feedback)	3.71	3.62	4.00
p-value	0.075	0.466	0.091

of training rounds they experienced, i.e., they just reported some percentage less but not really an accurate number. It also shows that the players would like to read less training stories for both groups.

It is interesting to note that the time at which they start feeling bored and the amount of training they are willing to do are closer to the 3 actual training rounds in group 2 than in group 1. This suggests that 3 might be about the limit of what people are willing to accept.

Table 11 shows the comparison of the players' response between group 1 and group 3. The players in group 1 who left more feedback feel more bored and want less training stories than the players in group 3 who left less feedback. The number of training stories after which the players started to feel bored is similar for the two groups.

To evaluate whether the number of training stories will influence the statistical properties of the story and option ratings, I compare the difference of the averages of the prefix/option ratings between the first two groups of players. Table 12 shows the averages of the prefix ratings and option ratings. As we can see from Table 12,

Table 12: The comparison of averages of prefix and option ratings for different numbers of training stories.

	Average prefix ratings	Average option ratings
Group 1 (5 training)	3.44	3.16
Group 2 (3 training)	3.67	3.39
p-value	0.378	0.206

collecting different numbers of training stories does not have a significant influence over the average of prefix ratings or option ratings.

In summary, all the players want to read less number of training stories and rate less number of prefixes/options, although the statistical property (average) of the ratings are not influenced by the amount of feedback collected, which suggests that the players do not change their rating behavior based on the amount of training and the rating quality is not reduced by collecting more feedback. The human study results are consistent with our expectation. In the next section, I will develop an algorithm based on active learning to reduce the amount of training data for the personalized drama manager.

5.1.2 Use Active Learning to Select Training Stories

In the current drama management system, the drama manager randomly picks story prefixes/options for the first few training stories for each new player (Section 4.6). The personalized DM uses the collected prefix/option ratings to bootstrap the player models. As suggested by the human studies in previous section, the players prefer less number of training stories in general. In this section, I aim to reduce the amount of training by choosing the initial story prefixes/options through active learning.

Active learning is a semi-supervised learning algorithm that can achieve higher accuracy with fewer training labels through interactively query an oracle (a human or some other information source) [59]. The queries posed by the active learner are usually in the form of unlabeled data instances to be labeled by the oracle. In terms of the interactive narrative, the data instances will be the story prefixes/options and the labels are their ratings. The personalized DM asks the new player for ratings of the story prefixes/options that are supposed to identify the player's type more efficiently than randomly chosen story prefixes/options. Using active learning, the personalized DM will be capable of creating a more accurate player model with less requirement for the ratings.

The active learning algorithm is used to improve the performance of the pPCA algorithm. In pPCA algorithm, the covariance matrix of the prefix rating vector $\mathrm{Var}(r)$ equals to $WW^T + \sigma^2 I$, where W is the conversion matrix in Equation 1, σ^2 is the variance of ϵ in Equation 1, and I is the identity matrix. The correlation matrix Corr for r can be computed by normalizing the covariance matrix $\mathrm{Var}(r)$. Then I define a utility score $U(p)$ for each story prefix p_i as follows:

$$U(p_i) = \sum_{j \in S_i} |Corr_{i,j}| \tag{18}$$

where S_i is the set of prefixes of which the ratings are known. For an unknown prefix p_i, the lower the $U(p_i)$ score, the higher the probability that the prefix will be selected. In other words, the active learning algorithm aims to select prefixes that have low correlation with the prefixes of which the ratings are already known.

I performed two studies, one of which is on human players' data and the other is on simulated players' data, to evaluate whether and to what extent the active learning can help build the player models. The following two sections will describe the two studies in details.

5.1.2.1 Active Learning Study on Human Data

I applied the active learning algorithm on the human players' data collected in Section 4.6. In total I have prefix ratings from 151 testing players. Each player read 6 stories, the first five of which are training stories from one branching story graph and

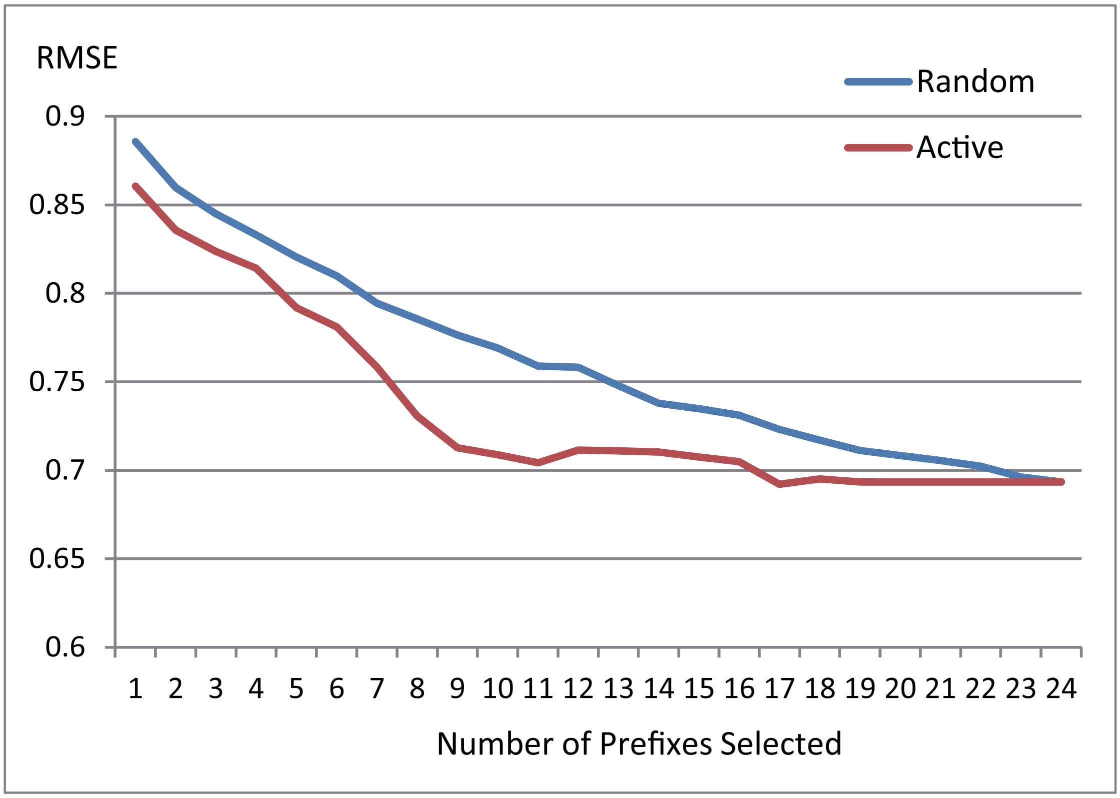

Figure 26: Use active learning to improve the story preference model on human players' prefix ratings.

the last one is the testing story from the other branching story graph. The prefix ratings of the first five stories are used to bootstrap the player model for each player.

I compared two prefix selection algorithms: random selection and active selection. For random selection, I randomly select a subset of prefix ratings from the first five stories for each player to bootstrap the player model. For active selection, I use active learning to select a subset of prefixes to bootstrap the player model. The player models are then used to predict the corresponding player's prefix ratings in the last story. Figure 26 shows the root mean square errors of the prediction on the last story for the two player models using different prefix selection strategies.

As shown in Figure 26, the player model built with actively selected ratings achieves better RMSE that randomly selected ratings. On average, active learning can reduce the number of training prefixes by 4.5. Thus using active learning can improve the bootstrapping speed of the player model for each new player in the personalized drama manager.

5.1.2.2 Active Learning Study on Simulated Data

To further understand how active learning can improve player modeling and compare different active prefix selection strategies in the personalized drama manager, I performed another group of study on simulate players. The simulated players are created using Robin's Laws in the same way as in Section 3.3.7 except that I remove the randomness in the simulated ratings for the purpose of better comparing different prefix selection strategies.

Each simulated player reads one to seven training stories and one testing story plot point by plot point. The drama manager collects their prefix and option ratings after every plot point. At each plot point in the training stories, the drama manager uses one of the following strategies to select target prefixes for the simulated players and guides the players to the selected target using the personalized guidance algorithm:

- *Random:* the drama manager randomly selects a successive plot point in the branching story graph.
- *Child:* the drama manager selects an immediate child plot point that has the lowest utility score which is defined in Section 5.1.2.
- *Offspring:* the drama manager selects a child plot point that leading to a prefix that has the lowest utility score in the current subtree.
- *Any:* the drama manager selects a prefix that has the lowest utility score in the entire branching story graph.

For each of the four strategies, I create 10,000 simulated players. Figure 27 shows the average root mean square errors of the DM prediction in the last story for different number of training stories. As shown in the figure, the two active selection strategies

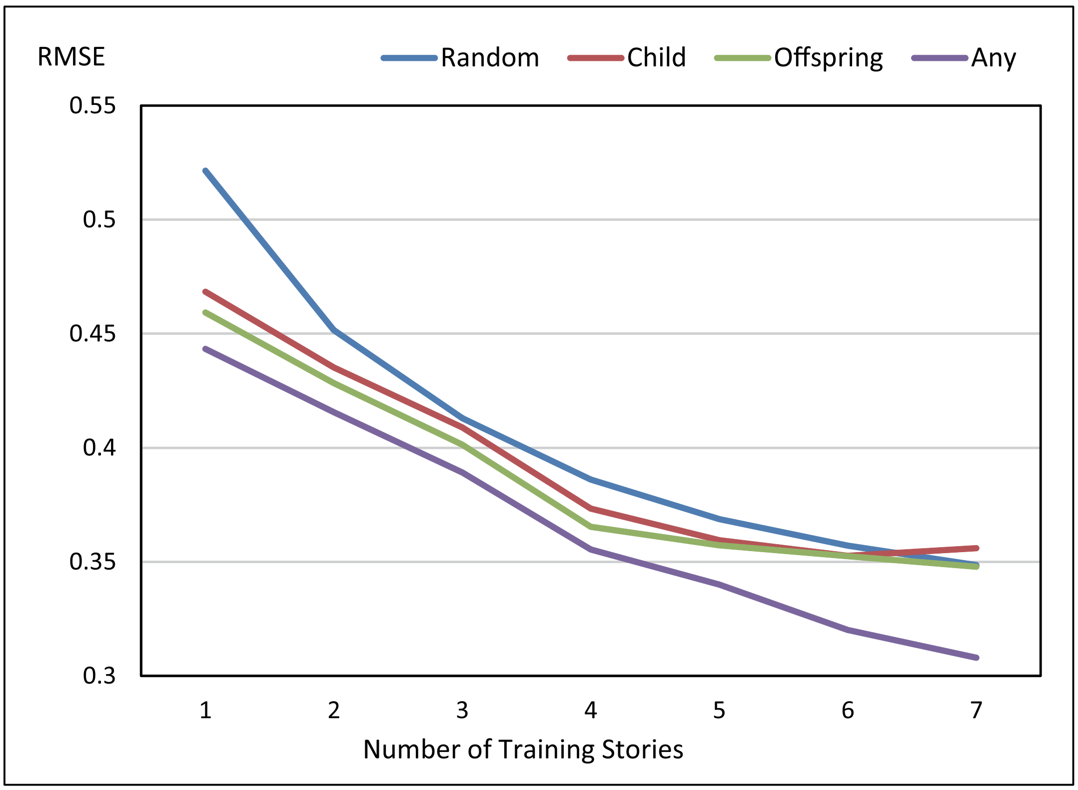

Figure 27: Use active learning to improve the story preference model on simulated players' prefix ratings.

(*Child* and *Offspring*) achieve lower root mean square errors than *Random* strategy. The *Any* strategy can be viewed as a theoretical minimum since it is not possible to use it with human players in real interactive narrative.

Both the experiment on human players' data and on simulated players show that active learning can help reduce the amount of training data required to build the story preference model. The human study results in Section 5.1.1 show that the players would like to read less number of training stories and rate less number of prefixes/options in the interactive storytelling. Thus the personalized drama manager using active learning to select training prefixes is capable of improving the players' experience in interactive storytelling.

5.2 Incorporate Author's Preference

Up until now, the personalized drama manager builds the story/option preference model based only on the players' preference. It will be beneficial to take the author's intention into consideration when building the player models. Incorporating the author's preference will not only increase the author satisfaction, but also help to build better player preference models for the personalized drama manager. Unfortunately, it is difficult to collect the story and option preference from the authors of the choose-your-own-adventure books. In this section, I will only discuss how the personalized drama manager can utilize the author's preference for different types of author preference.

In general, there are three different types of author preference over the full-length stories in the branching story graphs:

- *Uniform:* the author has the same preference over all the full-length stories.
- *Subset:* the author prefers towards a subset of stories but do not like other stories outside the subset.
- *Distribution:* the author has a distribution of preference ratings for the stories.

The first two types can be viewed as specialized types of the third type *Distribution*, which is the most general one. I intentionally separate the first two types from the third one for the purpose of using different strategies of incorporating the author's preference into the player models.

For the author preference type *Uniform*, the personalized drama manager does not need to make any change to the current player preference models because the uniform preference does not provide any additional information and can be just viewed as a noninformative prior.

For the preference type *Subset*, the author's preference for the full-length stories can be represented as a vector containing 0s and 1s, where 1 means in the subset and 0 means not in the subset. The personalized drama manager can utilize two strategies to incorporate the author's preference. The first strategy is to only select target full-length stories that are also in the author's subset, which is equivalent to multiplying the players' preference ratings by the author's preference. The second strategy is to use a weighted average of the players' preference ratings and the author's preference vector. The first strategy ensures that most selected stories will be within the subset that the author prefers, while the second strategy does not. Thus the first strategy leans more towards the author's preference.

For the third preference type *Distribution*, the author's preference can be represented as a preference vector on a scale of 1 to 5, which is similar to the players' preference vector for the story prefixes. The two strategies that the personalized DM uses for the preference type *Subset* can also be used here. In addition, the personalized drama manager can use the author's preference vector as a prior knowledge for the training of the player preference model as described in Chapter 3. The author's preference is then viewed as one type of players' preference in this case.

Incorporating the author's preference into the personalized drama manager increases author satisfaction, although the story preference ratings could decrease for players who have very different preference than the author. For the players who have similar preference to the author, incorporating the author's preference can not only build a better player preference model, but also reduce the required number of training stories and shorten the model bootstrapping phase for the players. Through achieving a balance between the author's preference and the players' preference, the personalized drama manager can potentially improve the players' experience, while increase the author's satisfaction.

5.3 Conclusion

In this chapter, I explore two possible improvements to the personalized drama manager: using active learning to reduce the required number of training stories and incorporating author's preference into the player preference models. The human studies on players' training experience show that the players would like to read less training stories and rate less number of prefixes/options in the interactive storytelling. Both the experiments on human players' data and on simulated players demonstrate that using active learning can effectively reduce the amount of training data required to build the story preference model. The personalized drama manager equipped with active learning can build a better player preference model with less training data, thus improve the players' experience in interactive storytelling. Incorporating the author's preference into the player models can not only increase author satisfaction, but also reduce the required number of training stories and improve the experience for players with similar preference types to the author.

CHAPTER VI

CONCLUSIONS AND FUTURE WORK

In this chapter, I will summarize this dissertation and highlight its contributions. Then I will discuss possible directions of future work for the personalized drama manager.

6.1 Summary

Although widely used in recommender systems to personalize users' experience, player modeling is still an emerging field of study in computer games, especially interactive narrative. Prevailing player modeling approaches in interactive narrative classify players according to well-defined player types and using pre-defined mappings between types and plot point selection rules. These approaches require human designers to pre-determine the meaningful player types, even though there is no clear evidence of links between player type models and story preferences.

In this dissertation, I develop a a data driven player modeling algorithm—prefix based collaborative filtering algorithm—to model players' preference over stories in interactive narrative. Unlike traditional collaborative filtering problem, the story preference modeling is a sequential recommendation problem, in which each subsequent recommendation is dependent on the entire sequence of prior recommendations due to the sequential natural of stories. The PBCF algorithm is developed to address the sequential recommendation problem. The drama manager using the PBCF algorithm can build flexible player preference models dynamically from players' feedback without rigid assumption on pre-defined player types. The PBCF algorithm is capable of learning player types dynamically from players' feedback and better capturing players' story preference. Both human study and simulated study results in Chapter 3 show that the PBCF algorithm can model the players' preference with high accuracy.

The PBCF algorithm can improve the players' preference ratings if the drama manager selects stories for the players in interactive narrative. To allow the players make choices by themselves, I develop a personalized drama manager algorithm that can maximize the players' expected story ratings while preserve the player agency. The personalized drama manager works in a multi-option branching story graph. It is capable of predicting an individual player's preference over the stories using PBCF and the player's preference over the options using CF, modeling the probability the player transitioning to successive plot points, selecting a target story experience that can maximize the player's expected preference ratings, and guiding the player to the selected story experience in an interactive narrative system. The human study results in Chapter 4 show that the personalized DM significantly increases the players' story experience ratings and guidance successful rate in the testbed interactive narrative built with CYOA stories.

I further propose two possible improvements to the personalized drama manager: using active learning to reduce the required number of training stories and incorporating author's preference into the player preference models. Both the experiments on human data and simulated data demonstrate that the personalized drama manager using active learning can effectively reduce the amount of training data required to build the story preference model, thus improve the players' experience in interactive storytelling. Incorporating the author's preference into the player models can not only increase author satisfaction, but also reduce the required number of training stories and improve the experience for players with similar preference types to the author.

6.2 Contributions and Potential Applications

In this dissertation, I made the following contributions:

- I propose a new type of recommendation problem—sequential recommendation, in which each subsequent recommendation is dependent on the entire sequence of prior recommendations. I develop a prefix based collaborative filtering algorithm to address the sequential recommendation problem. The prefix based collaborative filtering algorithm is a data-driven preference modeling algorithm that does not require any assumption on pre-defined player types.

- I develop a personalized guidance algorithm that can guide the players in a multi-option branching story graph while preserve player agency. The personalized guidance algorithm allows the players to make their own choices but significantly increases the probability that the players choose selected branches in interactive narrative.

- I build a personalized drama manager that uses PBCF algorithm to model the players' preference over story plot points, selects target story plot points that maximize the player's expected preference ratings, and guides the player to the selected plot point using the personalized guidance algorithm. Human study results show that the personalized drama manager can influence the players' choice selection and significantly improve the players' story preference ratings while preserve the player agency. I also demonstrate that active learning can be used to further improve players' experience in interactive narrative.

In addition to the contributions described above, the development of the personalized drama manager system also provides immediate benefits in supporting several potential applications, including computer games, intelligent tutoring systems and other domains or problems which have sequential nature.

The initial goal of developing the personalized drama management system was to improve soldiers' experience and performance for a training system. The tasks and events in the training system have a sequential nature. Thus PBCF algorithm can be easily adapted to model users' performance in the training systems, in which case the PBCF algorithm should model the users' training scores instead of preference ratings. The PBCF can also be applied in intelligent tutoring systems to improve users' performance and experience. In fact, the PBCF algorithm has already been successfully used in an edition of the CRYSTAL ISLAND narrative-centered learning environment [35].

The personalized drama manager can not only be used in computer games and tutoring systems, but also be applied to other systems where we would like to model users' preference and influence the users' choice selections. For example in a online shopping case, traditional collaborative filtering algorithms make one-shot recommendation for a product. Using the PBCF, the system can recommend a sequence of products for a customer based on his/her preference or previous purchase history. Then the personalized drama manager algorithm can be applied to influence the customer's choices if we have a selected product for the customer.

6.3 Future Work

In this section, I will discuss two major future directions: how to guide the players in the case of repeated visits to the same branching plot point, and how to further reduce the amount of feedback the drama manager needs to collect from the players through using other features to build the player models.

6.3.1 Repeated Visits to the Same Branching Point

Human study results in Section 4.6 show that the players's preference ratings are significantly increased in the testing phase (the last session), in which the players visit a new branching story graph for the first time. It is still unclear that whether and how much the personalized drama manager can improve the players' experience during repeated visits to the same branching plot point.

The original player preference models over the story prefixes should be recalibrated based on the players' new experience. Temporal based collaborative filtering approaches can be used to model the changes of the players' preference [23, 24, 81]. Furthermore, the story preference model should take into consideration the players' characteristics such as whether the players prefer to explore more in the branching story graph, at which branching points the players are more likely to change their decisions, etc.

The personalized guidance algorithm also needs to be updated during the repeated visits. Some options are already exposed to the players. The personalized drama manager needs to either update the preference model for the options, or build another option selection model to predict the probabilities that the player will select each option based on all the information, e.g. whether the players have read the options, the previous ratings for the options, the number of times the players have visited this branch, the options shown to the player during previous visits, etc.

6.3.2 Building Player Models Using Other Features

In Section 5.1.2, I use active learning to reduce the number the training stories required to build the player models. It will be beneficial to further reduce the requirement for training data.

The personalized drama manager can utilize other features to bootstrap the player models for the new players. The players' option selection behavior is an useful indication for their preference [67]. The personalized drama manager can utilize the selected options to help build the preference model. For more complex computer games, the players' gameplay features could also be used to build the player models. The gameplay features have been used to model the players' boredom, frustration and fun [71, 78]. Further research is needed to study how these gameplay features are correlated with the players' preference over the stories. It is a promising research direction to build the personalized drama manager purely on these features instead of the explicitly preference ratings.

6.4 Conclusion

Improving player experience is an important goal for the drama manager in interactive narrative systems. Although personalized drama management has not been well explored in the literature, I believe that building a personalized DM is essential to enhance the player experience in interactive narrative systems and computer games. The personalized drama manager presented in this dissertation builds data driven player models without pre-defined player types, and guides the players to better story experience while preserves the player agency. Its capabilities have been tested and proved in the human studies and evaluations in the dissertation. As such, I conclude the thesis statement, first proposed in Section 1.1 have been achieved. The personalized drama manager marks an important milestone for personalizing the players' experience in interactive narratives. Its development helps us to understand important problems and identify possible solutions in the scientific pursuit of using artificial intelligence to improve user experience in interactive narratives, computer games, intelligent tutoring systems, and other computer systems.

APPENDIX A

STORY LIBRARY USED IN THE HUMAN STUDIES

In this research, I transcribed totally four choose-your-own-adventure books to create all the branching story graphs for the story library: *The Abominable Snowman*, *The Lost Jewels of Nabooti*, *Space And Beyond*, and *Journey Under The Sea*. For the human study in Section 3.3, all the four branching story graphs are used but no option is shown to the players. For the human study in Section 4.6, I used the branching story graphs—*The Abominable Snowman* and *The Lost Jewels of Nabooti*. Two additional options are authored for each branch. In total, there are three options pointing to each successive plot point at every branching point in the two branching story graphs. In this appendix, I describe in details all the branching story graphs for the four choose-your-own-adventure books and all the options used in the human studies.

A.1 Four Branching Story Graphs

Figure 28 shows the branching story graph for *The Abominable Snowman*. Figure 29 shows the branching story graph for *The Lost Jewels of Nabooti*. Figure 30 and Figure 31 show the branching story graphs for *Space And Beyond*. Figure 32 and Figure 33 show the branching story graphs for *Journey Under The Sea*.

I transcribed the stories in the original choose-your-own-adventure books through removing branches that led to "sudden death" outcomes and splitting/merging some pages in the books to ensure that every full-length story contains exactly six plot points. The mappings between the plot points in my branching story graphs and the pages in the original books are shown in Table 13 to Table 16.

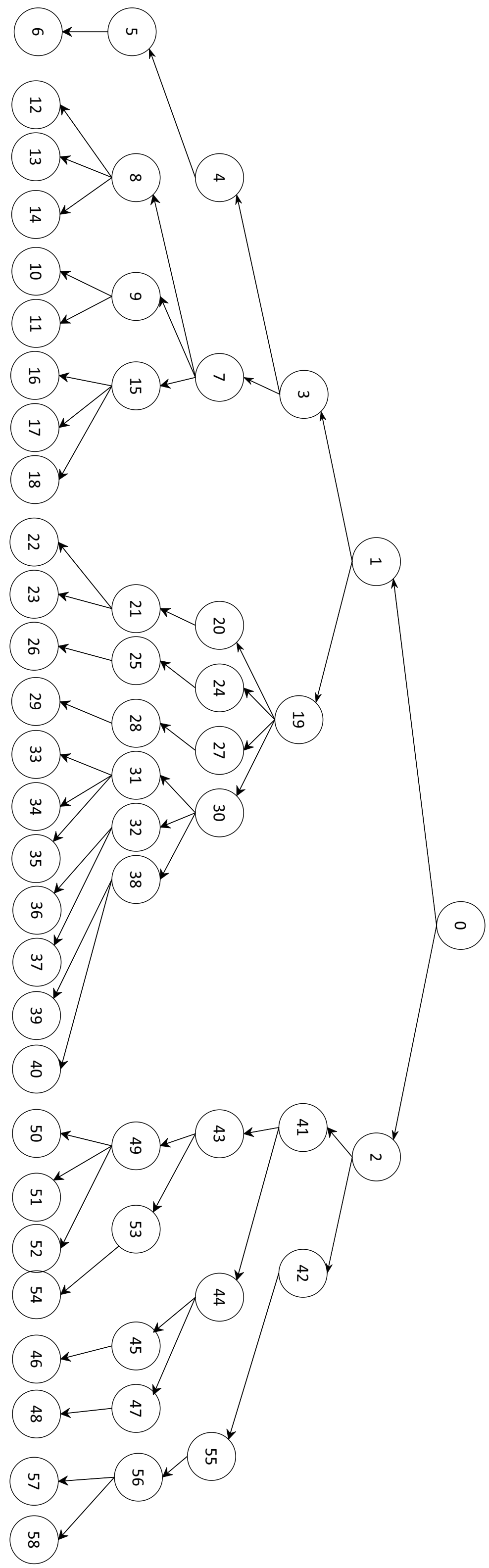

Figure 28: The branching story graph for the choose-your-own-adventure book: *The Abominable Snowman.*

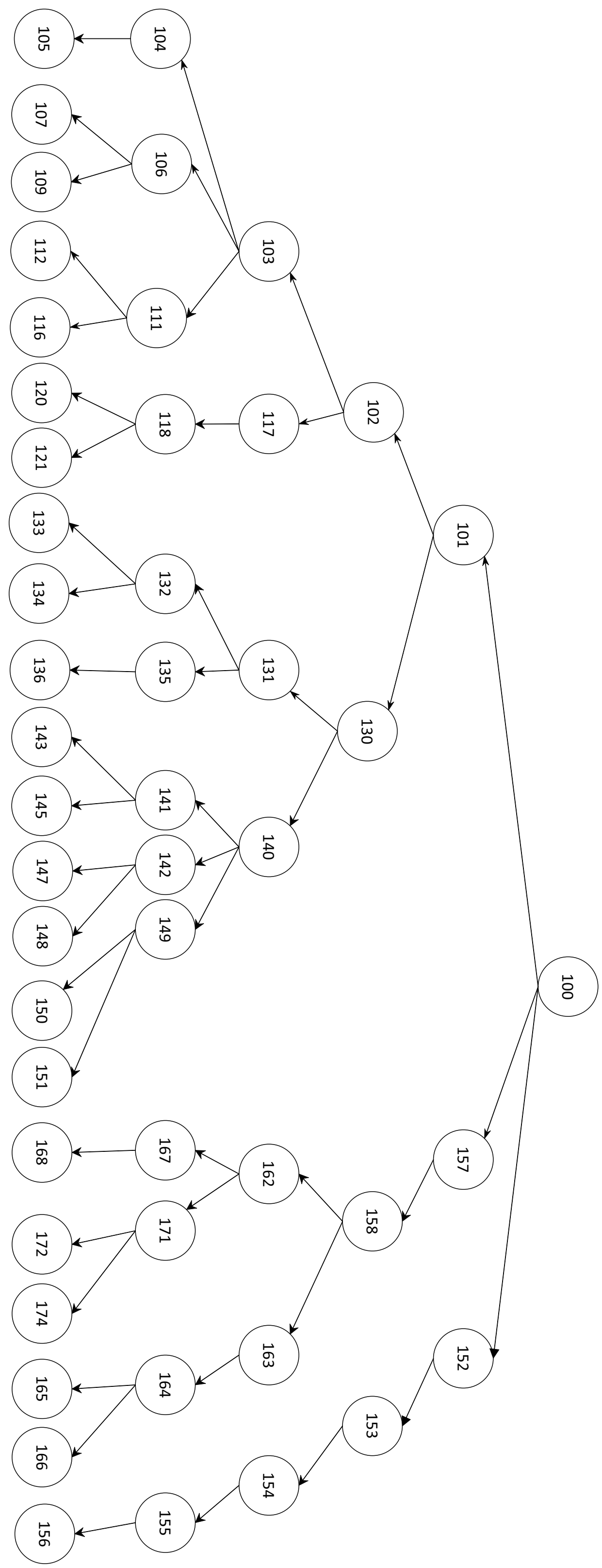

Figure 29: The branching story graph for the choose-your-own-adventure book: *The Lost Jewels of Nabooti*.

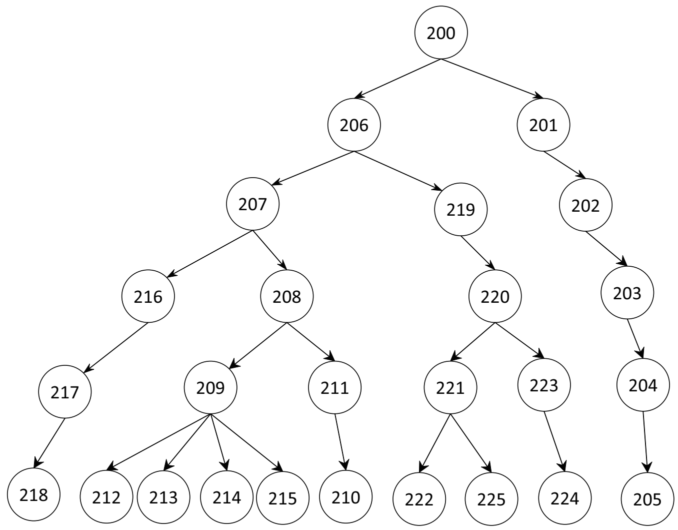

Figure 30: The first part of the branching story graph for the choose-your-own-adventure book: *Space And Beyond*.

Table 13: The plot point and page number mappings for *The Abominable Snowman*.

Plot Point	Page Number	Plot Point	Page Number	Plot Point	Page Number
0	1-5	1	7	2	8,10
3	9	4-6	15,32	7	20,31
8	43,58	9	43,62	10	82,96
11	82,98,109,60	12	77,91,101,103	13	77,95,106,111
14	80	15	45	16	55,76,78,100,104
17	55,76,78,102	18	57,75,73	19	13,14
20	22,34,21	21	46	22	59
23	64	24-26	22,34,21,47	27	23
28-29	33	30	23,38	31	50,65
32	50,67	33	86,37	34	86,114
35	87	36	85	37	83,99
38	48,49,68,69	39	88	40	89
41	16	42	19	43	24,26
44	27	45	39	46	51
47,48	42	49	40	50	63
51	51,70,92,97-107	52	72	53,54	116
55-57	28	58	29		

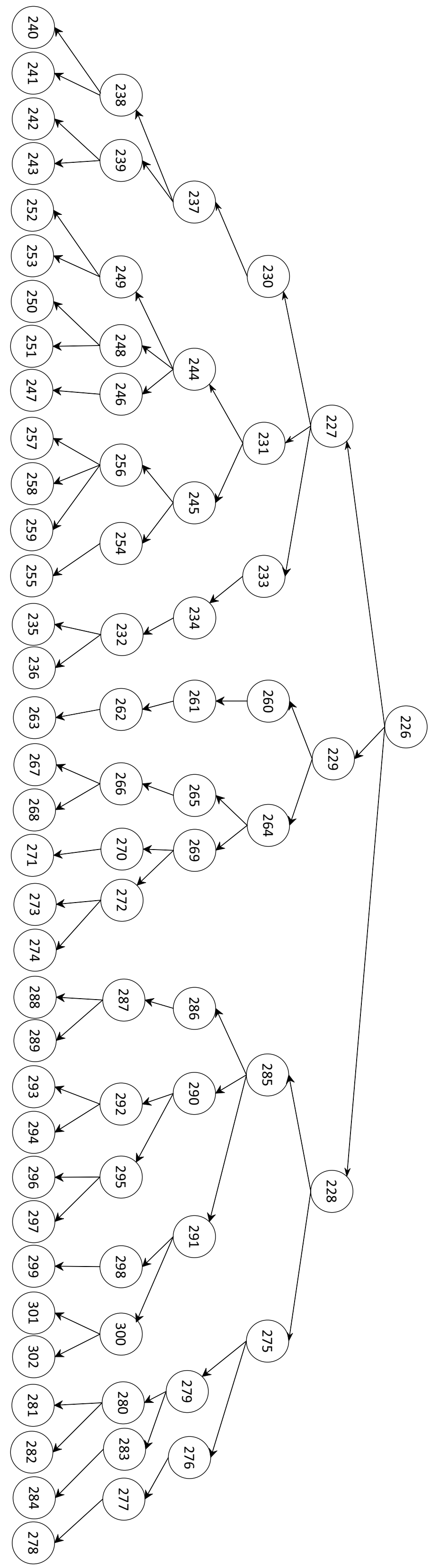

Figure 31: The second part of the branching story graph for the choose-your-own-adventure book: *Space And Beyond*.

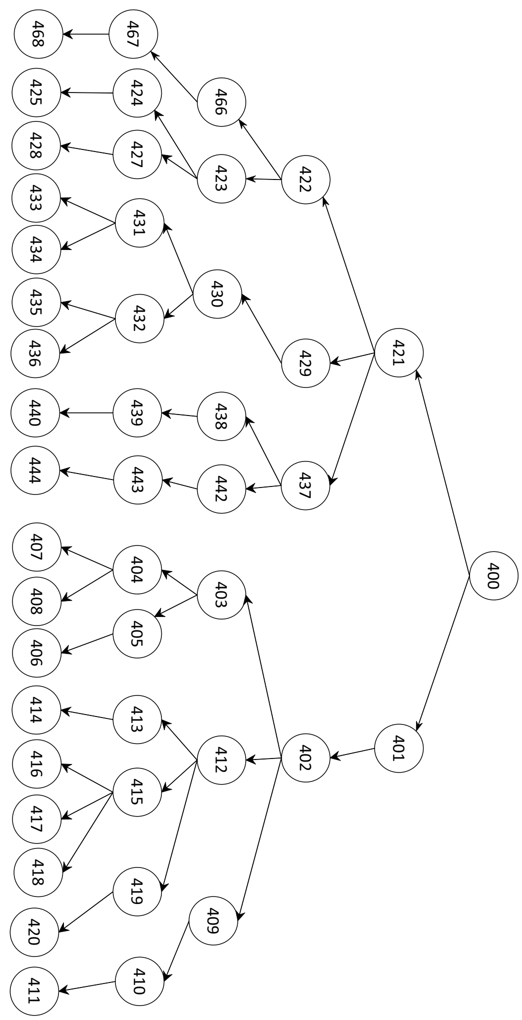

Figure 32: The first part of the branching story graph for the choose-your-own-adventure book: *Journey Under The Sea*.

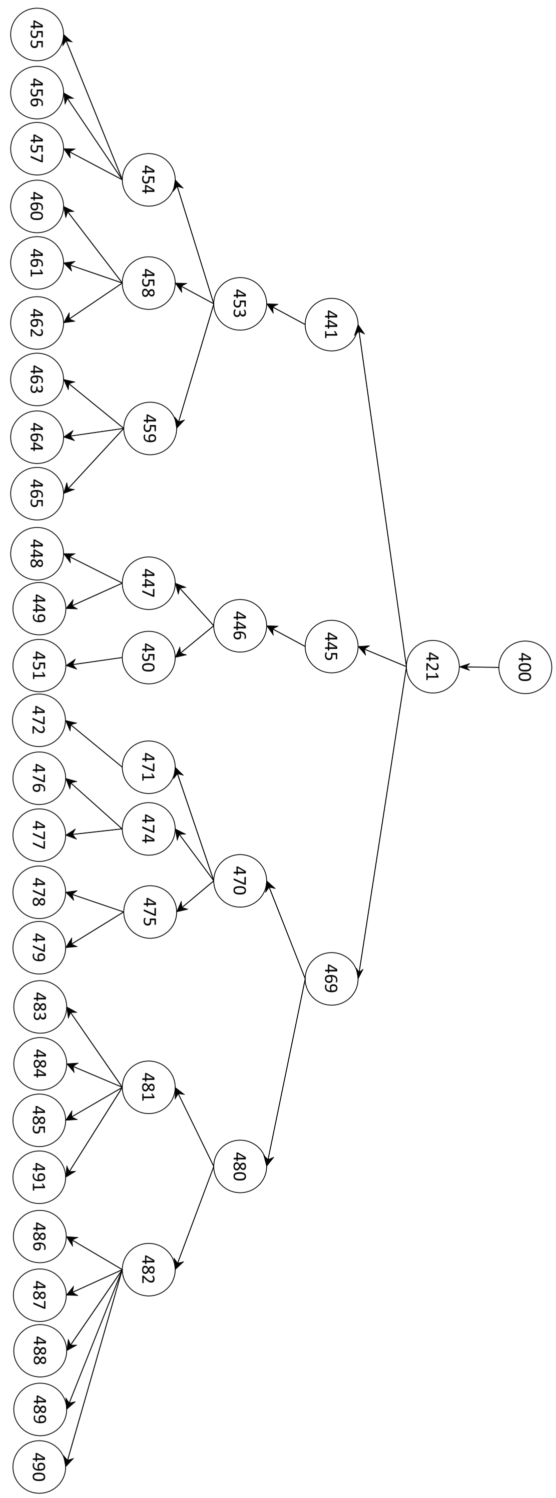

Figure 33: The second part of the branching story graph for the choose-your-own-adventure book: *Journey Under The Sea*.

Table 14: The plot point and page number mappings for *The Lost Jewels of Nabooti*.

Plot Point	Page Number	Plot Point	Page Number	Plot Point	Page Number
100	1-4	101	5	102	8
103	16	104	29	105	126
106	27,64,66	107	89	109	88,109,122
111	27,44,43	112	61,86,116	116	62,85
117	18,30	118	50	120	70,95
121	72,96	130	12	131	19,32,35
132	49	133	74	134	76
135	52	136	128	140	20,37,38
141	53,75	143	97	145	100,131
142	53,78	147	101,130	148	103
149	54	150	80	151	79
152-156	9,10,14,24	157,158	9,10,15	162	20,41
163	21	164	42	165	55,80
166	60	167,168	58	171	57,82
172	107	174	108,114		

A.2 Options in the Multi-Option Branching Story Graphs

The branching story graphs from the choose-your-own-adventure books *The Abominable Snowman* and *The Lost Jewels of Nabooti* are used in the interactive narrative system in Chapter 4. Two additional options are authored for each successive plot point at every branching point in the two branching story graphs. The following motivational theories, drawn from Petty [43] and Cialdini [9], are used to author the options:

- *Expert Source*: a desire to follow experts' opinions.
- *Scarcity*: a desire for something that will soon become unavailable.
- *Consistency*: a desire to appear consistent with what we have already done or said.
- *Social Proof*: a desire to imitate others in similar situations.
- *Reasoning*: a desire to follow arguments that sound rational.

Table 15: The plot point and page number mappings for *Space And Beyond*.

Plot Point	Page Number	Plot Point	Page Number	Plot Point	Page Number
200	1,2,4,5	201-205	9,10,22	206	8
207	19	208	42	209,210	60
211	56,59	212	82,106,112	213	82,106,113
214	82,107	215	83	216-218	39
219	20	220	21	221	41,43
222	62	225	57	223,224	44
225	57	226	1,2,3	227	6,12
228	7	229	6,14	230	25,47
231	25,26,45,46	232	66	233	25
234	47	235	84	236	85
237	64	238	86	239	87,90
240	94	241	114	242	115
243	116	244	67	245	70
246,247	92	248	89,108	249	89,109
250	117	251	120	252	118
253	119	254,255	94	256	91
257	110	258	111,121	259	111,122
260-263	32,50	264	32,49	265	69
266	69	267	95	268	96
269	71	270,271	97	272	98
273	124	274	125	275	15,16
276-278	31	279	34	280	51
281	72	282	74	283,284	52
285	17,36	286	55	287	77
288	104	289	105	290	54,73
291	54,78	292	99	293	126
294	127	295	101	296	128
297	129	298,299	102	300	103
301	131	302	130		

Table 16: The plot point and page number mappings for *Journey Under The Sea*.

Plot Point	Page Number	Plot Point	Page Number	Plot Point	Page Number
400	1,2	401	6,7	402	10
403	19	404	34	405,406	36
407	49	408	50,8	409,410	17
411	31	412	17,32	413	47
414	66	415	47,63	416	88
417	87,94	418	87,95,110	419,420	87,95,111
421	4	422	3	423	14
424,425	23	427,428	26,39	429	3,14,26
430	40	431	55	432	56
433	76	434	77	435	78
436	79	437	8	438-440	18,30
442-444	13,24	445	8,13	446	27
447	42	448	57	449	58
450,451	43	453	28,29	454	45
455	62	456	64,83	457	64,86
458	44,59	459	44,60	460	80
461	82,114	462	82,114	463	81,116
464	81,117	465	84	466-468	9,25,6
469	3,9,21	470	33	471	51
472	67	473	68	474	53,69
475	53,70	476	96	477	97
478	98	479	99,54	480	38
481	52	482	54	483	74,92
484	74,93,104	485	74,93,105	486	71,89,101
487	71,89,103	488	71,90,100	489	71,90,102
490	72	491	75		

- *Number of arguments*: a desire to follow statement that contains repetitive arguments expressed in different ways without new information.
- *Motivation–Friendship*: a desire for friendship.
- *Motivation–Safety*: a desire for being safe.
- *Motivation–Money*: a desire for being rich.
- *Motivation–Fame*: a desire for being famous.

Table 17 to Table 23 shows all the options, including the original ones and the new ones I authored, in the branching story graph in Figure 28. Table 24 to Table 31 shows all the options in the branching story graph in Figure 29. The column *From* contains the plot point numbers from which the options point. The column *To* contains the plot point numbers to which the option point. The column *Theory* shows which motivation theory is used to author the corresponding option.

Table 17: The options used in Figure 28.

From	To	Theory	Content
0	1	Original	You decide to expedition by yourself.
0	1	Friendship	You are worry about your friend Carlos. You decide to go to the Everest base to search for Carlos as soon as possible.
0	1	Consistency	You decide to follow your original plan and go directly to Kathmandu to search for the Yeti.
0	2	Original	You feel that Carlos is OK and want to meet with Mr. Runal.
0	2	Reasoning	It will be very helpful for your expedition if you can talk with a mountain expert before your trip. You decide to talk with Mr. Runal about you expedition plan.
0	2	Safety	This will be your first trip to Himalayas. It is much safer to meet with Mr. Runal, a mountain expert, for safety tips in mountain area.
1	3	Original	You and Runal want to search below the base camp in the valley.
1	3	Social	You and Runal decide to search the canyon where most previous reports of the Yeti were reported.
1	3	Safety	You decide to search downhill since it is safer to go down the trail and avoid the ice and snow up in the mountain.
1	19	Original	You and Runal want to search above the base camp.
1	19	Reasoning	Most of the previous reports about Yeti are outdated and probably misleading. That's why no one has ever gotten a Yeti photo. You decide to ignore those reports and search above the camp.
1	19	Original	Carlos's analysis was that he believed the Yeti lived high in the mountain. You believe your friend's judgment and decide to search uphill.
2	41	Friendship	Carlos could be in danger. You decide to go ahead to search for Carlos.
2	41	Fame	You continue the expedition for Yeti since you could be the first person in the world to get Yeti photos.
2	41	Consistency	You have been preparing for the expedition for such a long time that you decide to continue to the expedition.
2	42	Safety	It is wise to play it safe. You decide to postpone the expedition for the Yeti.
2	42	Money	You go for tigers since you can make a good fortune by selling photos of tigers in the Terai region.
2	42	Expert	Mr. Runal is an expedition expert. He recommends you to postpone the Yeti expedition. You decide to follow his advice.
3	4	Original	You decide to ignore the message.

Table 18: The options used in Figure 28 (Continued).

3	4	Friendship	Carlos might be in danger and need help. You decide to go to look for Carlos immediately.
3	4	Reasoning	Carlos went to the expedition alone. But the woman said there were two persons. It might not be your friend Carlos who left the message. You decide to investigate what happened.
3	7	Original	You obey the message.
3	7	Safety	The trail looks so dangerous. You do not know what will happen ahead. You decide to find a safe place to stay.
3	7	Arguments	Carlos is your friend whom you trust. Since he left a message asking you to go back to the base camp, you decide follow his advice.
4	5	Original	Continue.
5	6	Original	Continue.
7	8	Original	You decide to follow the Yeti with Runal.
7	8	Scarcity	How could you miss such a rare opportunity to get the photos of the Yeti? You and Runal grab a camera immediately and follow the sound before it is gone.
7	8	Expert	The woman lives local and seems to know a lot about Yeti. You and Runal decide to trust her and accept the invitation from Yeti.
7	9	Original	You decide to follow the Yeti alone.
7	9	Friendship	You come all the way to find Carlos. He might be in danger. The Yeti could lead you to him. You decide to follow Yeti but leave Runal behind as a rear guard for safety.
7	9	Money	You might earn a lot of money if you make new discovery about Yeti. But you do not want to share the money with Runal. You decide to run down the trail to follow the Yeti by yourself.
7	15	Original	You decide to return to the base camp and the helicopter.
7	15	Reasoning	There have been so many people in the world searching for the Yeti for such a long time. But still no people have found the Yeti yet. How could the women find it so quickly? You decide not to believe in her.
7	15	Safety	Judging from the noise, you feel that the Yeti were enraged. Something bad might happen. You decide that it is safer to stay away from it.
8	12	Original	You decide to retreat.
8	12	Safety	It does not look safe here. You decide to get out of here while you still can.
8	12	Arguments	The place is eerie. The red backpack might be taken from Carlos or he might leave it as a warning for you. You decide to stay there but keep silent.

Table 19: The options used in Figure 28 (Continued).

8	14	Original	You give the special bird call whistle that you and Carlos used as your emergency code.
8	14	Friendship	The red backpack might be taken from Carlos. Is he in trouble? You call his name to see if he is nearby.
8	14	Reasoning	The place is dark. The backpack indicates that Carlos might be nearby. But he might not be able to see you. You decide to shout out loudly to see if anyone is there.
9	10	Original	You hesitate to go into the door.
9	10	Expert	Mr. Runal knows more about this area. He told you to come back to him. You decide to go back to Mr. Runal.
9	10	Safety	It could be really dangerous inside the doors. You decide to turn back.
9	11	Original	You decide to go inside the door.
9	11	Fame	The doors might lead you to the Yeti. You could be the first person in the world to find the Yeti. You push the door.
9	11	Consistency	You do not want to retreat and give up at the last minute on your way searching for the Yeti. You decide to go in to the door.
15	16	Original	You decide to go back.
15	16	Safety	The big foot prints might be left by the Yeti. The giant foot prints indicate the Yeti are huge. It is too dangerous to follow the prints. You decide to talk with Runal for the next step.
15	16	Arguments	The helicopter is smashed. The rotor blades are twisted and the Plexiglas is shattered. You decide to stay out of there as soon as possible.
15	18	Original	You decide to follow the prints.
15	18	Consistency	The giant footprints looks like Yeti footprints. It has been your dream to find the Yeti. You will not quit half way through your expedition. You ask Runal to go follow the prints with you.
15	18	Scarcity	The footprints will be soon covered by snow. You might never get so close to getting photos of the Yeti ever again. You and Runal decide to follow the prints.
19	20	Original	You stay there and look around.
19	20	Safety	You are very concerned of the danger ahead along searching for Yeti. You decide to stay at Sangee's shop and purchase new safety equipment.
19	20	Money	It seems that you can buy some local stuff here and sell them later in the US for a profit. You decide to pick up some new things.
19	30	Original	You want to know more about Sangee.

Table 20: The options used in Figure 28 (Continued).

19	30	Reasoning	Sangee has very rich experience exploring the area close to Kathmandu. It will be very helpful for. You decide to talk with Sangee to know more about the Yeti.
19	30	Arguments	It has always been your dream to find Yeti. You have come a long way to search for it. Sangee is an expert on local expedition. You decide to invite Sangee to your expedition.
20	21	Original	Continue.
21	22	Original	You make up a fantastic story.
21	22	Safety	They will kill you if you tell them nothing. You decide to say something to earn yourself more time.
21	22	Money	They will heavily reward you if you provide helpful information. You decide to try to fool in hopes of reward.
21	23	Original	You insist that you know nothing.
21	23	Reasoning	They should know something about the map because they have it in their shop. You might get yourself into trouble if you lie to them. You decide to tell the truth.
21	23	Arguments	You saw the map by chance. But you do not know anything about the map. Although they may reward you if you can provide helpful information, you decide to explain yourself further.
24	25	Original	Continue.
25	26	Original	Continue.
27	28	Original	Continue.
28	29	Original	Continue.
30	31	Original	You hesitate where to go.
30	31	Expert	It seems difficult for you to make the decision. Sangee is very familiar with this area. You would like to talk with Sangee to know more about the Yeti.
30	31	Arguments	The Yeti can be near the Everest area, but they can also live in the Annapurna area. You need to think more about it.
30	32	Original	You decide to go to the Annapurna region.
30	32	Safety	It is too dangerous to go to the Everest region, the highest mountain on earth. Safety is most important for you. You decide to avoid the Everest region for safety.
30	32	Reasoning	The Everest is so high and so cold that it is unlikely for the Yeti to survive there. You decide to follow Sangee's advice.
30	38	Original	You decide to go to the Everest region.
30	38	Fame	You will be famous if you find the Yeti in the Everest area, the highest mountain in the world. You will go to the Everest area.

Table 21: The options used in Figure 28 (Continued).

30	38	Scarcity	You might never find another chance to visit the Everest area, the highest mountain in the world. You cannot miss it.
31	33	Original	You decide to take a picture of the creature.
31	33	Fame	You will be the first person in the world to find the Yeti. You will be famous all over the world. You decide to follow the Yeti with your camera.
31	33	Money	You will be richer than you can imagine if you can get the Yeti pictures. You decide to grab a camera.
31	35	Original	You decide not to take a picture of the creature.
31	35	Safety	It will be too dangerous to take a picture of the creature. You decide to grab something that can defend yourself.
31	35	Safety	The camera flash will attract the Yeti. You do not know what it will do to you. You decide to grab something that can frighten the creature.
32	36	Original	You want to respond to their help.
32	36	Scarcity	The flashlight might be from the Yeti. You do not know how long you have to wait for another chance. You cannot let it go away. You decide to go to take a look.
32	36	Arguments	You see a light flash on Annapurna. The flash repeats and seems to be a signal. Someone might be in trouble. You decide to go to help them.
32	37	Original	You decide to let Sangee return to Pokhara for help.
32	37	Safety	It will be too dangerous to walk below the glacier. You decide to let Sangee return to Pokhara for help.
32	37	Reasoning	It is dark and cold out there. You might not be very helpful by yourself. You decide to let Sangee return to Pokhara for help.
38	39	Original	You decide to take the risk.
38	39	Consistency	It has been your dream to search for the Yeti. There will be more opportunities in Khumbu Icefall. You decide to go there.
38	39	Expert	Sangee is very familiar with this area. He recommends you to start your search from Khumbu Icefall. You decide to follow his advice and go to the Khumbu Icefall.
38	40	Original	You can't decide it.
38	40	Safety	It will be really dangerous in the icefall. You could lose your life. You hesitate.
38	40	Arguments	Great pieces of ice tumble from the glacier and pile up like children's building blocks. The ice may crack and give way when you least suspect. Many have died in these icefalls. You need to think more about it.
41	43	Original	You accept Runal's offer to join you.

Table 22: The options used in Figure 28 (Continued).

41	43	Safety	It will be much safer if you two go together and look after each other. You need a companion in case of emergency. You decide to ask Runal to come with you.
41	43	Scarcity	It is a great chance to have an expedition expert on your team. If you hesitate the other expedition team will know your plan and offer higher compensation to Runal. You cannot miss the chance to have Runal on your team.
41	44	Original	You decline Runal's offer to join you.
41	44	Reasoning	Mr. Runal seems not to support your expedition from the very beginning. You doubt that he will really help you on your expedition. You decide to find an excuse to decline his offer.
41	44	Consistency	Carlos and you make a good climb team. It is not easy to get along with a new companion in a short time. You decide to go alone for Carlos.
42	55	Original	Continue.
43	49	Original	You are ready for the secret knowledge of the Yeti.
43	49	Consistency	You have been searching for Yeti for such a long time. You will not give up at the last minute. You decide to continue your journey no matter what will happen to you.
43	49	Fame	You will be famous if you become the first person in the world to find the Yeti. You decide to accept the offer for now, take some pictures of the Yeti and the go back home.
43	53	Original	You reject the offer of secret knowledge.
43	53	Reasoning	No one has ever published a Yeti photo before. It seems to be impossible to keep the secret of the Yeti for so long if so many people know about it. You decide not to trust them.
43	53	Arguments	Your life will be changed if you accept the offer. It sounds terrible that your life will never be the same forever. You decide to find an excuse to retreat.
44	45	Original	You try to make amends.
44	45	Safety	Mr. Runal seems to be angry. You do not know what he can do to sabotage your expedition. It is wise to do as he wishes.
44	45	Scarcity	It will be very helpful to have a mountain expert on your team. You change your mind and invite Mr. Runal to accompany you before he changes his mind to go expedition with the other team.
44	47	Original	You stick to your decision.
44	47	Consistency	You rejected Mr. Runal's offer to come with you. You will not change the decision you just made. You decide not to ignore him.

Table 23: The options used in Figure 28 (Continued).

44	47	Arguments	He does not support your expedition from the very beginning. He seems not to be friendly with you. You decide to get out of there as soon as possible.
45	46	Original	Continue.
47	48	Original	Continue.
49	50	Original	You are not prepared to change your life forever.
49	50	Arguments	You can quit right now. It is the last chance for you to quit this weird journey. You can never go back if you continue the journey. You decide to quit.
49	50	Safety	The whole thing is getting too weird. You do not know how dangerous it will be ahead in the journey. You decide to find an excuse to quit right now.
49	51	Original	You agree to take the journey.
49	51	Consistency	It has been your dream to find the Yeti. You will not give up at the last minute. You definitely agree to continue.
49	51	Fame	You will become world-famous to find the Yeti. You decide to accept their offer for the moment.
53	54	Original	Continue.
55	56	Original	Continue.
56	57	Original	You split up.
56	57	Arguments	You can cover more territory if you split up. You decide to split up.
56	57	Reasoning	Poachers may catch both of you if you stay together. You decide to split up so that if one of you get caught, the other one can go for help.
56	58	Original	You stay together.
56	58	Safety	It will be safer to stay together. You can look after each other. You decide to stay together.
56	58	Expert	Mr. Runal is an expedition expert. He suggests you stay together. You decide to follow his advice.

Table 24: The options used in Figure 29.

From	To	Theory	Content
100	101	Original	You agree to go on tomorrow's plane for Paris.
100	101	Friendship	Peter and Lucy are your cousins and best friends. You will go on tomorrow's plane for Paris to help them without hesitation.
100	101	Money	You will be much richer than you can imagine if you find the precious diamond. You go for Paris without hesitation.
100	157	Original	You want to help them but need more information.
100	157	Safety	It is safer to think carefully before you take any action. You decide to talk with them and get more information.
100	157	Arguments	You are still puzzled after reading the letter. Although you feel Peter and Lucy might be in danger, you hesitated about what to do next. You decide to talk with them first.
101	102	Original	You accept his offer.
101	102	Expert	The man is an expert on the Jewels of Nabooti. You decide to accept the offer and follow his advice.
101	102	Scarcity	You don't speak French and barely know anyone in France. You might not have another chance to find someone to help you. You accept his offer to search Nabooti together.
101	130	Original	You make excuses and refuse his help.
101	130	Safety	Looking at the man by appearance, it seems not safe to work together with this stranger. You refuse his offer politely by saying that you are expecting a friend to pick you up.
101	130	Reasoning	The coincidence seems to be too suspicious. You want to be very careful at each step and not let other people sabotage your plan. You make excuses and refuse his help.
102	103	Original	You take a seat on the left near the door.
102	103	Safety	You pick a seat close to the door since it will help you can escape easily if you need to.
102	103	Reasoning	Your companion looks suspicious. You still do not know his background or what he wants. You decide not to follow what he said and sit close to the door.
102	117	Original	You sit with your back to the wall.
102	117	Arguments	You should watch the doors and windows. There may be your enemies. You decide to sit with your back to the wall away from the door.

Table 25: The options used in Figure 29 (Continued).

102	117	Expert	You accompany seems to be very experienced in this situation. You decide to follow his advice and sit with your back to the wall.
103	104	Original	You grab a chair to defend yourself.
103	104	Scarcity	This will be your only chance to talk with Molotawa and know more about the jewel. You decide not to run away and you grab a chair to fight against him.
103	104	Consistency	You have not finished talking with Molotawa and are still confused and have a lot of questions for him. You want to continue your conversation and decide not to run away. You grab a chair to fight against him.
103	106	Original	You get out of the cafe and try to lose yourself into a crowd.
103	106	Reasoning	You know the entrance and exit of the cafe. You don't think the person would do harm on you in public. You race for the nearest exit and try to lose yourself into a crowd.
103	106	Safety	The man coming towards you is really strong. He is much stronger than you and he has a knife in his hand. It is safe to avoid him. You race for the nearest exit and try to lose yourself into a crowd.
103	111	Original	You get out of the cafe and wish someone can come to help you.
103	111	Safety	It is too dangerous. The man might kill you. You run out of cafe and wish someone can help you.
103	111	Consistency	You are here to find the jewel, not to create trouble. You decide to get out of here and continue searching for the jewel. You wish someone can help you continue searching the jewel.
104	105	Original	Continue.
106	107	Original	You decide to go to Morocco.
106	107	Reasoning	Molotawa used the word "gave" You believe he does not have the jewels and decide to go to Morocco instead.
106	107	Safety	It is not safe to go back to talk with Molotawa. Someone there tried to kill you. You decide to go to Morocco instead.
106	109	Original	You want to be back to talk with Molotawa.
106	109	Scarcity	Molotawa told you it would be your only chance to meet him. You might never find Molotawa again in the future. You decide to go back to talk with Molotawa.
106	109	Arguments	Molotawa is from the Nabooti group. He knows about the jewels. You decide to go back to talk with Molotawa to research more on the jewels.

Table 26: The options used in Figure 29 (Continued).

111	112	Original	You hesitate and then go peacefully.
111	112	Safety	It is too dangerous to fight with them! You decide to go out peacefully for safety.
111	112	Reasoning	You are alone with a pistol. They have more people and more weapons. There is no chance to fight with them. You decide to go peacefully.
111	116	Original	Fight it out.
111	116	Scarcity	They will not hurt you in the thirty-six seconds. You can attach them by surprise during the thirty-six seconds. After the thirty-six seconds, there will be no chance for you to fight it out. You decide to grab the last minute and fight with them.
111	116	Arguments	They shot at you. Now you have a pistol. You can fight back at them. You decide to fight.
117	118	Original	Continue.
118	120	Original	You decide to contact the Nabooti.
118	120	Reasoning	The car might be from the Nabooti. They seem to know you are here. You decide not to hide from them. You will contact the Nabooti directly.
118	120	Arguments	Dakar is a big city. It will take too long to search in Dakar. You decide to contact the Nabooti directly.
118	121	Original	You decide to search in Dakar.
118	121	Safety	It might expose yourself and put you in danger if you contact the Nabooti directly. You decide to search in Dakar first.
118	121	Expert	Ouobessa is a local police officer. You decide to follow his advice and search in Dakar first instead of directly contacting the Nabooti.
123	124	Original	Continue.
124	125	Original	You go to the Mountains of the Moon.
124	125	Arguments	The Mountains of the Moon are very beautiful. It bears the secret of your search. You decide to go there first.
124	125	Reasoning	You need to find Nabooti as quickly as possible. You are closer to the Mountains of the Moon. You decide to go to the Mountains of the Moon first.
124	128	Original	You travel to the headwaters of the Zaire River.
124	128	Safety	The Mountains of the Moon is one of the highest mountains in the world. It will be dangerous to climb it. You decide to start searching from the headwaters of the Zaire River.
124	128	Expert	The man recommends starting searching from the headwaters of the Zaire River. He seems to know everything in your mind. You follow his advice to go the river.

Table 27: The options used in Figure 29 (Continued).

125	126	Original	You ignore the warning and go into the hut.
125	126	Consistency	You come along all the way to search for the jewels. You will not give up at the end. You decide to ignore the warning.
125	126	Money	The warning and the curse mean something precious might be inside the hut. It could be the jewels you are looking for. You definitely go into the hut.
125	127	Original	You heed Ouobessa and refuse to enter.
125	127	Safety	It means it is death to enter. You will not put your life in danger.
125	127	Arguments	There are three chicken bones and two dead mice across the doorway. Ouobessa tells you that it is magic. It is a curse and a warning. You refuse to enter.
128	129	Original	Continue.
130	131	Original	You go to the jet.
130	131	Money	They will heavily reward you if you cooperate with them and find the jewels. You decide to go to the jet.
130	131	Expert	They are expert on the Jewels of Nabooti and might help you find the jewels. You follow their advice and go to the jet.
130	140	Original	You tell them that there must be some mistake.
130	140	Safety	It is too dangerous to go to the jet without knowing the background of these guys. You decide to make an excuse.
130	140	Reasoning	The dagger shows you a message. They seem to be not friendly with you. You decide to make an excuse.
131	132	Original	You give the ivory piece to the blind woman.
131	132	Safety	Your life is threatened by the driver. The woman might be able to save you. You decide to give the ivory piece to the woman.
131	132	Reasoning	Whoever sees the ivory will help you. It seems that the woman was trying to signal you and she might help you find the jewels. You decide to give the ivory piece to the woman.
131	135	Original	You decide to not do anything.
131	135	Arguments	The woman is so disgusting. And she is blind. You decide to avoid her as soon as possible.
132	133	Original	You decide to obey him.
132	133	Safety	They do not look nice to you and may hurt you if you do not obey them. You decide to obey him for the moment.

Table 28: The options used in Figure 29 (Continued).

132	133	Reasoning	Nobody has put anything in your pocket. There will be no harm for you to let them check your pocket. You decide to obey him.
132	134	Original	You decide to run for the door.
132	134	Arguments	You dont know what is going on. The old man wants you to give the jewels to him. You do not want to give the jewels to him. You decide to run for the door.
132	134	Consistency	You has been searching for the jewels for such a long time. You will not give the jewels to them. You decide to run for the door.
135	136	Original	Continue.
137	138	Original	Continue.
138	139	Original	Continue.
140	141	Original	You accept the offer of police protection and take a gun with you.
140	141	Arguments	You want to find the Jewels of Nabooti. The police offer to help you by providing you with a detective and a gun. They could be very helpful. You happily accept the offer of detective and gun.
140	141	Scarcity	It is very rare for police to offer a gun. You immediately accept the offer of police protection and the gun before they change their minds.
140	142	Original	You accept the offer of police protection but will not bring a gun.
140	142	Expert	The police are expert on handling these problem. You listen to the polices advice to bring a detective. But guns can cause trouble and you could hut yourself. You accept the offer to come with a detective but not the gun.
140	142	Safety	It will be much safer if the police comes with you. But guns can cause trouble and you could hurt yourself. You accept the offer of police protection but not the gun.
140	149	Original	You travel to Morocco alone, but accept the special phone number.
140	149	Reasoning	The police thought you were a smuggler. Now they offer to help you. It seems that what the police really want is to keep you under surveillance. You decide to travel to Morocco alone, but accept the special phone number.
140	149	Friendship	The police might take the jewels away if you find them. You need the jewels to save your cousins. You decide to travel to Morocco alone, but accept the special phone number.
140	149	Consistency	You only accept the special phone number since you always want to search for the jewels by yourself.

Table 29: The options used in Figure 29 (Continued).

141	143	Original	You go to meet the African People's Federation.
141	143	Expert	Raoul is a special detective and is more experienced in such circumstance. You follow his advice to meet with the African Peoples Federation.
141	143	Scarcity	It will be difficult to find all the members of African Peoples Federation in the future. Most of them will leave after the meeting. You decide to meet with them immediately.
141	145	Original	You skip the proposed meeting and push on for Morocco.
141	145	Reasoning	You do not want too many people to know about your plan of searching for the jewels. You do not know if any of them might sabotage you plan. You decide to skip the meeting and push on for Morocco directly.
141	145	Friendship	Peter and Lucy are in danger. You do not have too much time to meet with the African Peoples Federation. You decide to push on for Morocco directly.
142	147	Original	You follow the instructions in the note.
142	147	Expert	As an experienced detective, Raoul knows what he is doing. You follow the instructions in the note.
142	147	Arguments	The note seems to be left by Raoul. He asks you to leave by the back door and turn left at Rue Pelican, right at Rue Fugere and wait until you are contacted. You decide to follow the instructions.
142	148	Original	You wait for Raoul to return.
142	148	Reasoning	You are not sure if the note is left by Raoul. The note could be left by someone else to mislead you. You do not know their intention. You decide to wait in the restaurant and see what will happen.
142	148	Safety	A lot of weird things happened on your way to search for the jewels. It is unsafe to walk around in an unfamiliar area. You decide to wait in the restaurant.
149	150	Original	You go on in.
149	150	Scarcity	The door will close again very soon. You do not know when it will open again. It might never open again. You go inside the door.
149	150	Money	The jewels might be there. You could be very rich if you go inside. You decide to go on in.
149	151	Original	You ring the bell again and wait.
149	151	Safety	It could be very dangerous to go inside the door directly. You do not know what will happen if you go into the door directly. You decide to ring the bell again and wait.
149	151	Reasoning	The door is open but no one is there. It is weird. It could be a trap. You decide to ring the bell again and wait to see what will happen.

Table 30: The options used in Figure 29 (Continued).

152	153	Original	Continue.
153	154	Original	Continue.
154	155	Original	Continue.
155	156	Original	Continue.
157	158	Original	Continue.
158	162	Original	You decide to leave for Paris without getting the aid of the police.
158	162	Friendship	You worry about Peter and Lucy very much. It is all in your mind to help them find the jewels and save their lives as soon as possible. There is no time for hesitation. You decide to leave for Paris immediately without calling the police about the shooting.
158	162	Money	The more those people threatened you, the more valuable you believe the jewels are. But the police might take away the jewels after you find them. You decided to leave for Paris without contacting the police.
158	163	Original	You decide to get in touch with the Police first.
158	163	Reasoning	Police have professional experience to handle the threatening and cold calls. They might help you to identify the callers and find useful clues for searching for the jewels. You decide to get in touch with the Police.
158	163	Safety	It is too dangerous to solve the problem on your own. You decide to get in touch with the Police.
159	160	Original	Continue.
160	161	Original	Continue.
162	167	Original	You decide to skip Paris and head directly for Morocco.
162	167	Safety	The men standing in the check-in counter look dangerous. It is better to avoid them. You decide to play it safe and go for Morocco instead.
162	167	Arguments	There seemed to be a car following you to the airport. And there are three squarely built men with crew cuts standing by the counter watching the door. All of these look weird. You decide to change your plan to go to Morocco instead of Paris.
162	171	Original	You go up to Paris according to Peter and Lucy's plan.
162	171	Consistency	Starting the search from Paris has been your plan from the very beginning. You decide to stick to the plan.
162	171	Reasoning	The jewels were stolen in Paris. There might be some clues there. You decide to go to Paris according to Peter and Lucys plan.
163	164	Original	Continue.
164	165	Original	You agree to meet him in Morocco.

Table 31: The options used in Figure 29 (Continued).

164	165	Arguments	Anson wants to meet you in Morocco. He does not want to meet in Paris. You decide to meet him in Morocco.
164	165	Friendship	Anson is your good friend. He agrees to help you and wants to meet you in Morocco. You respect your friends desires and decide to meet him in Morocco.
164	166	Original	You pressure him to meet you in Paris.
164	166	Consistency	You, Peter and Lucy planned to meet in Paris. You decide to stick to your plan and pressure Anson to meet you in Paris.
164	166	Reasoning	The jewels were stolen in Paris. There should be some clues there. You decide to pressure Anson to meet you in Paris.
167	168	Original	Continue.
169	170	Original	Continue.
171	172	Original	You keep on going up the steps of the plane.
171	172	Safety	It will be too dangerous to run away. You keep on going up the steps of the plane.
171	172	Arguments	You captor points you with a gun. He does not allow you to run away. You keep on going up the steps of the plane.
171	174	Original	You run for it.
171	174	Scarcity	This might be your only chance to escape. You cannot miss the only opportunity to get away. You decide to run for it.
171	174	Reasoning	It will be much more difficult for you to escape after you board the plane. This will be your last chance to escape before boarding. You definitely run for it.

REFERENCES

[1] "http://en.wikipedia.org/wiki/choose_your_own_adventure," 3.3.1

[2] "http://en.wikipedia.org/wiki/expectation%e2%80%93maximization_algorithm," 3.2.2

[3] Aristotle, *The Poetics (Original work published in 350 B.C.E.)*. Buffalo, NY: Prometheus Books, T. Buckley translation, 1992. 2.1

[4] Bates, J., "Virtual reality, art, and entertainment," *Presence: The Journal of Tele-operators and Virtual Environments*, vol. 1, no. 1, pp. 133–138, 1992. 1

[5] Bates, J., "Virtual reality, art, and entertainment," *The Journal of Teleoperators and Virtual Environments*, 1992. 2.3.1

[6] Bhat, S., Roberts, D. L., Nelson, M. J., Isbell, C. L., and Mateas, M., "A globally optimal algorithm for TTD-MDPs," *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems*, 2007. 2.3.1, 3.1

[7] Bishop, C. M., *Pattern Recognition and Machine Learning*. Springer, 2006. 4.4.2.1, 4.4.2.1, 4.4.2.1

[8] Bruner, J., "The narrative construction of reality," *Critical Inquiry*, vol. 18, pp. 1–21, 1991. 1

[9] Cialdini, R. B., *Influence: The Psychology of Persuasion (Collins Business Essentials)*. Harper Collins Publishers, 2006. 4.1, A.2

[10] Demasi, P. and Cruz, A., "Online coevolution for action games.," *Proceedings of The 3rd International Conference on Intelligent Games And Simulation*, 2002. 2.4.1

[11] El-Nasr, M. S., "Interactive narrative architecture based on filmmaking theory," *International Journal on Intelligent Games and Simulation*, vol. 3, no. 1, 2004. 3

[12] El-Nasr, M. S., "Engagement, interaction, and drama creating an engaging interactive narrative using performance arts theories," *Interactions Studies*, vol. 8, no. 2, pp. 209–240, 2007. 1, 2.4.2

[13] Fairclough, C. R., "Story games and the opiate system.," *PhD thesis, University of Dublin - Trinity College*, 2006. 2.3.3

[14] Freytag, G., *The Technique of the Drama: An Exposition of Dramatic Composition and Art*. Johnston Reprint Corporation, 1968. 2.1

[15] Ghazanfar, M. A., Prügel-Bennett, A., and Szedmak, S., "Kernel-mapping recommender system algorithms," *Information Sciences*, vol. 208, pp. 81–104, 2012. 2.5.1

[16] Grisham, T., "Metaphor, poetry, storytelling and cross-cultural leadership," *Management Decision*, vol. 44, pp. 486–503, 2006. 1

[17] Hartley, T. and Mehdi, Q., "Online action adaptation in interactive computer games.," *Computers in Entertainment*, 2009. 2.4.1

[18] Huh, Y. E., Vosgerau, J., and Morewedge, C. K., "Social defaults: Observed choices become choice defaults," *Journal of Consumer Research*, vol. 41, no. 3, pp. 746–760, 2014. 1

[19] Hunicke, R. and Chapman, V., "AI for dynamic difficulty adjustment in games.," *Challenges in Game Artificial Intelligence AAAI Workshop*, 2004. 2.4.1

[20] Jennings-Teats, M., Smith, G., and Wardrip-Fruin, N., "Polymorph: Dynamic difficulty adjustment through level generation.," *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*, 2010. 2.4.1

[21] Johnston, J. R., "Narratives: Twenty-five years later," *Topics in Language Disorders*, vol. 28, no. 2, pp. 93–98, 2008. 1

[22] Kazmi, S. and Palmer, I. J., "Action recognition for support of adaptive gameplay: A case study of a first person shooter.," *International Journal of Computer Games Technology*, 2010. 2.4.1

[23] Kolda, T. G. and Bader, B. W., "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009. 6.3.1

[24] Koren, Y. and Bell, R., "Advances in collaborative filtering," in *Recommender systems handbook*, pp. 145–186, Springer, 2011. 6.3.1

[25] Lamstein, A. and Mateas, M., "Search-based drama management," *Proceedings of the 2004 AAAI Workshop on Challenges in Game Artificial Intelligence*, 2004. 2.2

[26] Lee, D. D. and Seung, H. S., "Algorithms for non-negative matrix factorization," *Advances in Neural Information Processing Systems*, vol. 13, pp. 556–562, 2001. 3.2.1, 3.2.2.2

[27] Leondar, B., "Hatching plots: Genesis of storymaking," *The Arts and Cognition*, pp. 172–191, 1977. 1

[28] Li, B., Lee-Urban, S., Appling, D. S., and Riedl, M. O., "Automatically learning to tell stories about social situations from the crowd," *Proceedings of the LREC 2012 Workshop on Computational Models of Narrative*, 2012. 2.2

[29] Lopes, R. and Bidarra, R., "Adaptivity challenges in games and simulations: A survey," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, no. 2, pp. 85–99, 2011. 1, 2.4

[30] Magerko, B., "Story representation and interactive drama.," in *AIIDE*, pp. 87–92, 2005. 2.3.4, 2.4.2

[31] Magerko, B. and Laird, J. E., "Mediating the tension between plot and interaction," *AAAI Workshop Series: Challenges in Game Artificial Intelligence*, 2004. 1, 2.3.4, 2.4.2

[32] Magerko, B., Stensrud, B. S., and Holt, L. S., "Bringing the schoolhouse inside the box - a tool for engaging, individualized training.," *Proceedings of The 25th Army Science Conference*, 2006. 2.4.1

[33] Mateas, M. and Stern, A., "Integrating plot, character and natural language processing in the interactive drama facade," *Proceedings of the 1st International Conference on technologies for Interactive Digital Storytelling and Entertainment*, 2003. 1, 2.3.3

[34] Medler, B., "Using recommendation systems to adapt gameplay," *International Journal of Gaming and Computer Mediated Simulations*, vol. 1, no. 3, pp. 68–80, 2008. 2.4.1, 2.5.2

[35] Min, W., Rowe, J. P., Mott, B. W., and Lester, J. C., "Personalizing embedded assessment sequences in narrative-centered learning environments: A collaborative filtering approach," *Proceedings of the Sixteenth International Conference on Artificial Intelligence in Education*, 2013. 6.2

[36] Mott, B. W. and Lester, J. C., "U-director: a decision-theoretic narrative planning architecture for storytelling environments.," *Proceedings of the Fifth International Conference on Autonomous Agents and Multi-Agent Systems*, 2006. 2.3.3

[37] Murray, J. H., "Hamlet on the holodeck - the future of narrative in cyberspace," *MIT Press*, 1997. 1

[38] Murray, J. H., *Hamlet on the Holodeck: The Future of Narrative in Cyberspace*. The MIT Press, 1998. 2.3.4, 4.6.4

[39] Nelson, M. J. and Mateas, M., "Search-based drama management in the interactive fiction Anchorhead.," *Proceedings of the First Artificial Intelligence and Interactive Digital Entertainment Conference*, 2005. (document), 1, 1.1, 2.2, 2.2, 6, 2.3.1, 2.3.1, 2.3.4

[40] Nelson, M. J., Mateas, M., Roberts, D. L., and Isbell, C. L., "Declarative drama management in the interactive fiction anchorhead," *IEEE Computer Graphics and Applications*, 2006. 2.3.1

[41] Peinado, F. and Gervas, P., "Transferring game mastering laws to interactive digital storytelling," *Proceedings of the 2nd International Conference on Technologies for Interactive Digital Storytelling and Entertainment*, 2004. 1, 2.4.2, 3, 3.3.6

[42] Pennington, N. and Hastie, R., "The story model for juror decision making," *Inside the Juror: The Psychology of Juror Decision Making*, pp. 192–221, 1993. 1

[43] Petty, R. E. and Cacioppo, J. T., "The elaboration likelihood model of persuasion," *Advances in experimental social psychology*, 1986. 4.1, A.2

[44] Platt, J. C., "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in Large Margin Classifiers*, pp. 61–74, 1999. 4.4.2.1

[45] Porteous, J. and Cavazza, M., "Controlling narrative generation with planning trajectories: the role of constraints," *Proceedings of 2nd International Conference on Interactive Digital Storytelling*, 2009. 2.3.2

[46] Prince, G., *A dictionary of narratology*. Lincoln, NE: University of Nebraska Press, 1987. 1, 2.1

[47] Rasmussen, C. E. and Williams, C. K. I., *Gaussian Processes for Machine Learning*. The MIT Press, 2006. 3.2.2.1

[48] Riedl, M., Saretto, C., and Young, R. M., "Managing interaction between users and agents in a multi-agent storytelling environment.," *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multi Agent Systems*, 2003. 1.1, 2.2, 2.3.2, 2.3.4

[49] Riedl, M. and Young, R. M., "From linear story generation to branching story graphs," *IEEE Journal of Computer Graphics and Animation*, vol. 26, no. 3, pp. 23–31, 2006. 1, 2.2, 2.2, 2.2

[50] Riedl, M. O., "Narrative generation: Balancing plot and character," *Ph.D. Dissertation, Department of Computer Science, North Carolina State University*, 2004. 1

[51] Riedl, M. O. and Bulitko, V., "Interactive narrative: An intelligent systems approach," *AI Magazine*, vol. 34, no. 1, 2013. 1, 2.1

[52] Riedl, M. O., Stern, A., Dini, D. M., and Alderman., J. M., "Dynamic experience management in virtual worlds for entertainment, education, and training," *International Transactions on Systems Science and Applications*, 2008. 1, 1.1, 2.3.2

[53] Roberts, D. L., "Computational techniques for reasoning about and shaping player experiences in interactive narratives," *Ph.D. Dissertation, School of Interactive Computing, Georgia Institute of Technology*, 2010. 4.6.4

[54] Roberts, D. L. and Isbell, C. L., "Desiderata for managers of interactive experiences: A survey of recent advances in drama management," *Proceedings of the First Workshop on Agent-Based Systems for Human Learning and Entertainment*, 2007. 2.3.1

[55] Roberts, D. L., Nelson, M. J., Isbell, C. L., Mateas, M., and Littman, M. L., "Targeting specific distributions of trajectories in MDPs," *Proceedings of the Twenty-First National Conference on Artificial Intelligence*, 2006. 1, 2.2, 2.3.1, 3.1

[56] Roberts, D. L., Riedl, M. O., and Isbell, C. L., "Beyond adversarial: The case for game AI as storytelling," *Proceedings of DiGRA 2009 Conference*, 2009. 1.1

[57] Rollings, A. and Adams, E., "Andrew rollings and ernest adams on game design," *New Riders Press*, 2003. 1

[58] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J., "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*, pp. 285–295, ACM, 2001. 2.5.1

[59] Settles, B., "Active learning literature survey," *Computer Sciences Technical Report 1648, University of Wisconsin-Madison*, 2010. 5.1.2

[60] Shaker, N., Yannakakis, G., and Togelius, J., "Towards automatic personalized content generation for platform games," *Proceedings of the 6th Annual Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2010. 2.4.1

[61] Sharma, M., Mehta, M., Ontanon, S., and Ram, A., "Player modeling evaluation for interactive fiction," *Proceedings of the Third Artificial Intelligence and Interactive Digital Entertainment conference*, 2007. 2.4.2, 3.3.4

[62] Sharma, M., Ontanon, S., Mehta, M., and Ram, A., "Drama management and player modeling for interactive fiction games," *Computational Intelligence*, 2010. 1

[63] Spronck, P., Ponsen, M., and Postma, E., "Adaptive game ai with dynamic scripting.," *Machine Learning*, 2006. 2.4.1

[64] Su, X. and Khoshgoftaar, T. M., "A survey of collaborative filtering techniques," *Advances in Artificial Intelligence*, 2009. 1.1, 2.5.1, 3, 3.2, 3.4

[65] Sullivan, A., Mateas, M., and N.Wardrip-Fruin, "Rules of engagement: Moving beyond combat-based quests," *Proceedings of Foundations of Digital Games, Intelligent Narrative Technologies III Workshop*, 2010. 2.4.1

[66] Sutton, R. S., "Temporal credit assignment in reinforcement learning," *Electronic Doctoral Dissertations for UMass Amherst, Paper AAI8410337*, 1984. 3.1

[67] Thue, D., Bulitko, V., Spetch, M., and Wasylishen, E., "Interactive storytelling: A player modelling approach," *Proceedings of the third Artificial Intelligence and Interactive Digital Entertainment Conference*, 2007. 1, 2.4.2, 3, 3.2, 3.3.6, 3.3.7, 2, 6.3.2

[68] Tipping, M. E. and Bishop, C. M., "Probabilistic principal component analysis," *Journal of the Royal Statistical Society*, vol. B61, no. 3, pp. 611–622, 1999. 3.2.1

[69] Togelius, J., Nardi, R. D., and Lucas, S. M., "Towards automatic personalised content creation for racing games.," *Computational Intelligence and Games*, 2007. 2.4.1

[70] Weyhrauch, P. W., "Guiding interactive drama," *Ph.D. Dissertation*, vol. School of Computer Science, Carnegie Mellon University, Pittsburgh, PA. Technical Report CMU-CS-97-109, 1997. 1, 2.2, 2.2, 2.3.1

[71] Yannakakis, G. N. and Togelius, J., "Experience-driven procedural content generation," *Affective Computing, IEEE Transactions on*, vol. 2, no. 3, pp. 147–161, 2011. 6.3.2

[72] Young, R. M. and Riedl, M. O., "Towards an architecture for intelligent control of narrative in interactive virtual worlds," *Proceedings of the 8th International Conference on Intelligent User Interfaces*, 2003. 2.3.2, 2.3.4

[73] Young, R. M., Riedl, M. O., Branly, M., Martin, R., and Saretto, C., "An architecture for integrating plan-based behavior generation with interactive game environments," *Journal of Game Development*, vol. 1, 2004. 1.1, 2.2, 2.3.2, 2.3.4

[74] Yu, H. and Riedl, M. O., "A sequential recommendation approach for interactive personalized story generation," *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, 2012. 1, 1, 2.2, 2.4.2

[75] Yu, H. and Riedl, M. O., "Data-driven personalized drama management," *Proceedings of the 9th AAAI Conference on Artificial Intelligence for Interactive Digital Entertainment*, 2013. 1

[76] Yu, H. and Riedl, M. O., "Toward personalized guidance in interactive narratives," *Proceedings of the 8th International Conference on the Foundations of Digital Games*, 2013. 1

[77] Yu, H. and Riedl, M. O., "Personalized interactive narratives generation: Sequential recommendation of plot points," *Transactions on Computational Intelligence and AI in Games (T-CIAIG)*, 2014. 1, 2.4.2

[78] Yu, H. and Trawick, T., "Personalized procedural content generation to minimize frustration & boredom based on ranking algorithm," *Proceedings of the 7th Annual Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2011. 6.3.2

[79] Yu, K., Zhu, S., Lafferty, J., and Gong, Y., "Fast nonparametric matrix factorization for large-scale collaborative filtering," *Proceedings of the 32nd SIGIR conference*, 2009. 3.2, 3.2.2.1

[80] Zhang, S., Wang, W., Ford, J., and Makedon, F., "Learning from incomplete ratings using non-negative matrix factorization," *Proceedings of the 6th SIAM Conference on Data Mining*, 2006. 3.2.1

[81] Zook, A. E. and Riedl, M. O., "A temporal data-driven player model for dynamic difficulty adjustment," *Proceedings of the 8th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2012. 2.4.1, 2.5.2, 6.3.1