Proceedings, The Eleventh AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE-15)

Optimizing Players' Expected Enjoyment in Interactive Stories

Hong Yu and Mark O. Riedl
School of Interactive Computing, Georgia Institute of Technology
85 Fifth Street NW, Atlanta, GA 30308
{hong.yu; riedl}@cc.gatech.edu

Abstract

In interactive storytelling systems and other story-based computer games, a drama manager is a background agent that aims to bring about an enjoyable and coherent experience for the players. In this paper, we present a personalized drama manager that increases a player's expected enjoyment without removing player agency. Our personalized drama manager models a player's preference using data-driven techniques, predicts the probability the player transitioning to different story experiences, selects an objective experience that can maximize the player's expected enjoyment, and guides the player to the selected story experience. Human study results show that our drama manager can significantly increase players' enjoyment ratings in an interactive storytelling testbed, compared to drama managers in previous research.

Introduction

Storytelling, in oral, visual, or written forms, plays a central role in various types of media, including novels, movies, and televisions. An interactive narrative is a new type of storytelling 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 (Riedl and Bulitko 2013). Compared to traditional storytelling systems, the interactive narrative gives the players the opportunity to change the direction or outcome of the stories, thus increasing player engagement. There are many ways to achieve interactive narrative. A simple technique is to construct a *branching story graph*—a directed acyclic graph in which nodes contain narrative content (e.g., plot points) and arcs denote alternative choices of action that the player can choose. Branching story graphs are found in the choose your own adventure novels, and also used to great effect in hypermedia and interactive systems.

More sophisticated interactive storytelling systems often employ a *Drama Manager* (DM), an omniscient background agent that monitors the fictional world and determines what will happen next in the player's story experience, often through coordinating and/or instructing virtual characters in response to player actions (Bates 1992). The goal of the DM is to increase the likelihood that a player will experience an enjoyable and coherent narrative. In prevailing interactive storytelling systems, the human game designers usually describe in high or low level what a "good" story should be. A DM then works to increase the likelihood that players will have narrative experiences that satisfy the descriptions given by the game designers (Nelson and Mateas 2005; Weyhrauch 1997; Roberts et al. 2006; Riedl et al. 2008; Magerko and Laird 2005; Mateas and Stern 2003). In other words, the DMs are surrogates for the game designers.

We believe that DMs should factor player preference into their decisions on how to manipulate the narrative experience (Thue et al. 2007; Yu and Riedl 2012). A DM that can optimize players' perceived experience thus is *also* a surrogate for the game players. Thue et al. (2007) create an interactive storytelling that models players' preference using fixed player types. In our previous research, we used a collaborative filtering player modeling algorithm to learn players' preferences over trajectories through a branching story graph without pre-defined player types (Yu and Riedl 2012). We previously proposed a graph modification algorithm to manipulate the likelihood of players following certain trajectories (Yu and Riedl 2013a; 2014).

Our previous drama management system works on branching story graphs where multiple options can point to the same plot point. The evaluation showed that the technique can influence the selection of a subsequent plot point, but did not incorporate the player preference model. In subsequent studies, reported in this paper, we show that the prior technique does not significantly increase player preference ratings for complete story experiences.

In this paper, we build off our previous work and present a new DM that uses the previous player modeling algorithm but maximizes players' *expected* enjoyment. The personalized DM algorithm presented in this paper chooses a successive branch that simultaneously increases the players' enjoyment and the probability of the player selecting the branch at every branching point. Our evaluation shows that the new technique outperforms earlier techniques *and* significantly increases players' ratings for their experiences.

Background and Related Work

Drama management has been widely used in interactive storytelling systems to guide the players through a story ex-

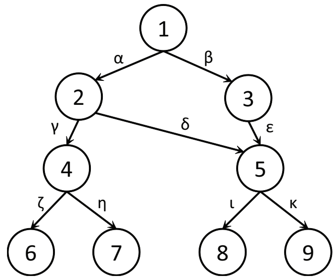

Figure 1: A simple branching story graph.

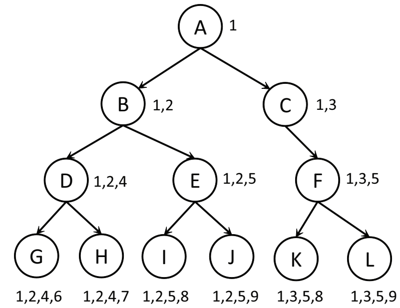

Figure 2: The prefix tree converted from the branching story graph in Figure 1.

perience pre-defined by game designers (Riedl and Bulitko 2013). Most of these Drama Management techniques do not consider players' preference and move the story forward in a way partially or completely conceived by a human designer.

Previous personalized DMs learn the player model using pre-defined discrete player types. PaSSAGE (Thue et al. 2007) builds the player model using Robin's Laws five game player types: Fighters, Power Gamers, Tacticians, Storytellers, and Method Actors. PaSSAGE models each player as a five dimensional vector and learns the vector through observations of the player's behavior in a CYOA style story world. Similar dimensional player models are found in Peinado and Gervás (2004) and Seif El-Nasr (2007).

In our previous research, we presented a data-driven player modeling algorithm that modeled player preferences over story experience in a branching story graph—Prefix-Based Collaborative Filtering (PBCF) (Yu and Riedl 2012). The PBCF algorithm is a data-driven technique that makes no pre-defined dimensional assumption and uses collaborative filtering to predict players' preference ratings for successive trajectories in the branching story graph. We further proposed a DM to increase the probability the player choosing selected plot points (Yu and Riedl 2013b; 2013a). The DM used a *multi-option branching story graph* that could have multiple options pointing to the same child plot point. It selected a subset of options to maximize the probability the player choosing the intended plot point selected by the DM. However, we did not implement a fully functional personalized DM agent that used the PBCF or other preference models to predict players' preference. Instead, our previous DM randomly selected a successive plot point as its target at each branching point in the multi-option branching story graph. We demonstrate in this paper that our previous DM will not perform well even using the PBCF player modeling algorithm because the DM may fail to guide a player at some branching points, leading the player to a subgraph where there is no appropriate plot point for the current player. In this paper, we present a new personalized DM that uses the PBCF algorithm and a new DM algorithm to maximize the *expected* player enjoyment in interactive narrative systems.

As in our previous work, the personalized DM assumes that an interactive narrative experience can be represented as a *branching story graph*. Figure 1 shows a simple branching story graph, in which nodes (denoted by numbers) represent plot points and arcs (denoted by Greek letters) represent alternative choices that players can choose. A full-length story is a path through the graph starting at the root node and terminating at a leaf node. While the representation is simple, many other drama management plot representations are reducible to the branching story graphs (Yu and Riedl 2012).

Prefix-Based Collaborative Filtering

The prefix-based collaborative filtering uses collaborative filtering to learn players' preference over story plot point sequences (Yu and Riedl 2012). Collaborative filtering algorithms are capable of detecting patterns in users' ratings, discovering latent user types, and predicting ratings for new users. Due to the sequential nature of stories, a player's preference over a plot point depends on the history of plot points the player has visited.

The PBCF extends standard CF algorithms to solve the *sequential* recommendation problems. The PBCF works in a prefix tree that is generated from the branching story graph. Each node in the prefix graph incorporates all the previous experienced plot points in the corresponding branching story graph. The children of a prefix node are those prefixes that can directly follow the parent prefix. Figure 2 shows a prefix tree that is converted from the branching story graph in Figure 1. Given the prefix tree representation, the PBCF uses collaborative filtering algorithms to learn and predict players' preference ratings over the prefix nodes. Notice that throughout the paper, we will use numbers to represent plot points, uppercase letters to represent prefixes, and Greek letters to represent options.

The PBCF algorithm can predict a player's preference over the story prefixes and select a successive plot point best for the current player. A DM is required to influence the player's decisions and maximize enjoyment.

Multi-Option Branching Story Graph

To increase the probability that the player transitions to the selected plot points, we proposed a variation of the branching story graph—*multi-option branching story graph*—in which multiple options could point to the same plot point (Yu and Riedl 2013a). An *option* is a CYOA style choice that the players can select in the multi-option branching story graph. Figure 3 shows top three layers of the multi-option branching story graph converted from Figure 1.

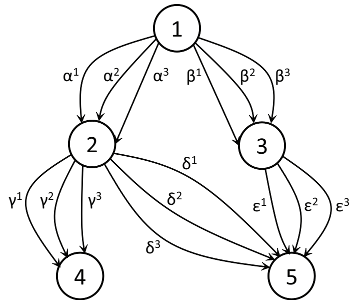

Figure 3: Example of a multi-option branching story graph.

The personalized DM uses collaborative filtering algorithms to additionally model the players' preference over the *options*. Given a desired child plot point that can lead to the optimal full-length story experience (a leaf node in the prefix tree) selected by the PBCF, a personalized DM can pick a particular subset of the options to present to the player such that at least one option leads to each child. This ensures true player agency and also increases the likelihood that the player will pick the option that transitions to the desired child plot point.

Our previous personalized DM selects an objective full-length story based only on the PBCF algorithm. It does not consider the probability that the player transitions to the selected full-length story. Thus it is possible for the player to transition to a subtree where there is no preferred full-length story for the player. For example, assume that the PBCF predicts that a player's preferences over the leaf nodes G, H, I, J, K, and L in Figure 2 are $4, 4, 4, 4, 1$, and 5, respectively. Their personalized DM will attempt to guide the player to the node L. Let's further assume that after the DM intervention, the current player still has a much higher probability to choose the option that transitions to prefix node K instead of L at 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, implemented in this paper, is to select a full-length story from G, H, I, or J as the objective when the player is at node A.

Personalized Drama Manager

In this section, we describe our new personalized DM algorithm. The personalized DM uses the PBCF algorithm to model players' preference over the story trajectories and works in the multi-option branching story graph.

Our personalized DM approach is summarized as follows. First, for a particular player, the personalized DM models his/her preference for all the possible trajectories using the PBCF algorithm. Second, the personalized DM uses standard CF to model the player's preference for all the options in the multi-option branching story graph. Third, the personalized DM models the probability that the player reaches each full-length story experience. Finally, the personalized DM chooses an objective full-length story that maximizes the expected enjoyment for the current player and selects a subset of options to maximize the probability the player transitioning to the objective full-length story.

Option Preference Modeling

We create the multi-option branching story graph through authoring multiple options between all the plot points and their immediate successors using a variety of motivational theories (Yu and Riedl 2013a). Collaborative filtering algorithms are used to model the players' preferences over the options. We have players rate the options presented after each plot point in a training phase and construct an option-rating matrix which is similar to the product rating matrix in traditional CF algorithms. Non-negative Matrix Factorization (NMF) (Lee and Seung 2001; Zhang et al. 2006) and probabilistic PCA (pPCA) (Tipping and Bishop 1999) are used to model the option preference ratings and predict option ratings for new players.

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 (Bishop 2006) 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:

$$\mathbb{E}[y_i|\mathbf{X}_i] = \mathrm{Logit}(\boldsymbol{\theta}' \cdot \mathbf{X}_i) \tag{1}$$

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

The Probit regression model (Bishop 2006) is similar to Logit regression, except that the Logit function is substituted with a Gaussian cumulative distribution function in Equation 1. The probabilistic SVM (Platt 1999) 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, we 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:

$$(\max_{\alpha\in\mathbb{O}^I_J}\{\mathrm{R}(\alpha)\},\ \min_{\beta\in\mathbb{O}^I_K}\{\mathrm{R}(\beta)\})' \tag{2}$$

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:

$$\mathrm{P}^I_{J,K} = \mathrm{f}(\mathbf{x}^I_{J,K}; \boldsymbol{\theta}) \tag{3}$$

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 (Bishop 2006).

For example, suppose a player is at prefix node A in Figure 2 (plot point 1 of the branching story graph) and the DM selects node C (plot point 3) as the objective for the player. The DM has six options to select from as in Figure 3. 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 prefix node I, we define $\mathbb{P}^I_L$ to be the probability that the player transitions to a leaf prefix 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 prefix tree of Figure 2, suppose the player is at the root node A. The probability that the player transitions to node L: $\mathbb{P}^A_L = \mathrm{P}^A_{C,B} * \mathrm{P}^F_{L,K}$.

Objective Full-length Story Selection

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

$$L^* = \mathrm{argmax}_{L_i \in \mathrm{Leaf}^I} \{\mathrm{R}(L_i) * \mathbb{P}^I_{L_i}\} \tag{4}$$

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 previous section.

Personalized Drama Manager Algorithm

Our personalized DM puts all the models to use as follows. For a new player, the personalized DM 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 prefix tree, the personalized DM uses the algorithm in Figure 4 to guide the player.

Notice that it is not strictly necessary to collect story and option ratings as in step 7. We do it in our 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 predictions in step 2 and 3. On the other hand, if we do not collect new ratings, it will not be necessary for the personalized DM to re-predict the ratings for full-length stories and options after every plot point.

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 4
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 next prefix node based on the player's choice
10: **end while**

Figure 4: The personalized drama manager algorithm.

Evaluation

To evaluate our personalized DM, we conducted a group of human studies in an interactive storytelling system built with choose your own adventure stories. We hypothesize that our personalized DM will be better at increasing players' enjoyment in the interactive storytelling system, as compared to baseline DMs. In this section, we will describe the story library and the interactive storytelling system we built, the training and testing of the personalized DM, human study results and discussions.

Story Library and System Setup

We built the story library using two choose your own adventure books: *The Abominable Snowman* and *The Lost Jewels of Nabooti*, were transcribed into two branching story graphs. We modified the stories such that each possible narrative trajectory contains exactly six plot points. On average each full-length story contains around 1,000 English words. 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 we have 134 story prefix nodes in the two trees.

We authored two additional options for each branch in the two branching story graphs as in (Yu and Riedl 2013a). In the final multi-option branching story graphs, there are three different options per successor plot point at every branching point. We have totally 275 options in the two multi-option branching story graphs.

In the human study, 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 bigger rating number indicates a higher preference. We created our storytelling system using an open source tool Undum (http://undum.com/). Figure 5 shows a

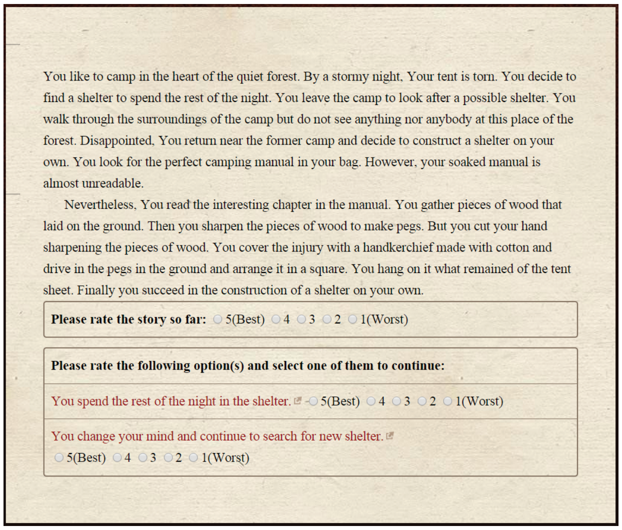

Figure 5: A screenshot of the interactive storytelling testbed.

screenshot of our online interactive storytelling system. The figure shows two plot points, a place for players to rate the story-so-far, and two options. The human study is composed of two phases: model training and testing, which will be described in the following sections.

Training the Personalized DM

We recruited 80 participants from Amazon's Mechanical Turk (MT). 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 we had 410 valid playthroughs 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. We 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. We 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, we 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 rest 10% of ratings in the prefix-rating matrix. The process is repeated 50 times. The best average root-mean-square-error (RMSE) for pPCA algorithm is 0.576 (dimension 46), and for NMF algorithm is 0.743 (dimension 12). Thus pPCA is used to model players' story preference in the testing phase.

To train the option preference model, we randomly select 80% of the training players to learn a option preference CF model. For the rest 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. We repeated the process for 50 times. The best average RMSE for pPCA algorithm is 0.550 (dimension 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.

We train the branch transition probability model using the predicted option ratings from the learned option preference model and the players' option selection. Similar to option preference model learning, we randomly select 80% of the training players to learn a option preference CF model. For the rest 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 average prediction accuracies for the Logit, Probit, and probabilistic SVM algorithm are 78.89%, 78.19%, and 79.35%. We 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.

Testing the Personalized DM

We recruited another 101 players, divided into three groups as described below, from MT 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 increase the player's enjoyment.

Personalized DM Algorithm Comparison For the purpose of comparison, we have implemented the following three guidance algorithms for the personalized DM:

- *HighestRating (HR)*: at each node in the prefix tree, the personalized DM selects an target full-length story based on predicted ratings of the stories. This is exactly the same as our previous DM (Yu and Riedl 2013a).
- *HighestMeanRating (HMR)*: at each node in the prefix tree, the personalized DM selects one 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 2. 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)*: this is our new personalized DM algorithm as in Figure 4.

In the human study, the above three personalized DM algorithms use the same PBCF story preference model and option preference model for the purpose of comparison. At

Table 1: The comparison of the three guidance algorithms.

Algorithm	w/o_DM	with_DM	p-value	Successful 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%

each branching point, the personalized DM 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.

The 101 testing players are assigned to the three groups: 28 players for *HR*, 26 players for *HMR*, and 47 players for*HER*. Table 1 shows the results of the comparison of the three algorithms. The first column (w/o_DM) and the second column (with_DM) show the average full-length story ratings for stories that are without DM guidance (average ratings in the first five trials) and with DM guidance (average ratings in the sixth trial). The Wilcoxon signed-rank test is used to compare the ratings for w/o_DM stories and with_DM stories. The p-values are shown in the third column. The last column shows the percent of the time the players chose the options transitioning to the desired plot points selected by the DM. As we can see from Table 1, the personalized DM algorithm *HMR* and *HER* can significantly increase the players' preference rating for their story experience. The *HER* algorithm has a much higher guidance successful rate than the other two algorithms. We recruited more players for the *HER* algorithm in order to compare to the equal-number-of-option case in the next section. In fact the with_DM ratings for *HER* was 4.13 ($p<0.001$) after we recruited only 24 players.

We further compared the players' ratings for with_DM stories under the three different DM algorithms. The results show that the with_DM ratings for the *HER* algorithm are significantly higher than the *HR* algorithm ($p=0.037$). The with_DM 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).

Select One Option Per Branch In the above human studies, the personalized DM picked two options for the desired branch but only one option for all the other successive branches. We also studied whether the personalized DM would perform differently if it picked equal number of options for each successive branch. We recruited another 50 players from Mechanic Turk. The study was conducted as in the above testing process. The only difference was that the personalized DM 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 w/o_DM and with_DM are 3.28 and 3.74, respectively. The with_DM ratings are significantly higher than the w/o_DM ratings ($p=0.004$). The average guidance successful rate is 70.8% for all the 50 players. Thus the personalized DM with the *HER* algorithm can also increase the players' preference ratings significantly when the DM picks one option for each successive branch.

Discussion

The Logistic model is capable of correctly predicting the players' branch transitions for 78.9% of the time. Although the more complicated non-linear probabilistic SVM can achieve higher predicting accuracy on the training data, the generalization error will probably not be reduced due to the prediction error in the option ratings. In the future, we will 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 DM's decision process, our personalized DM significantly increases the players' enjoyment in the interactive storytelling system. Our DM algorithm *HER* beats both *HR* and *HMR* in terms of the players' enjoyment ratings. The guidance successful rate of *HER* is also greatly improved against *HR* and *HMR* since our DM does not select objectives that the players have low chance to reach. The with_DM rating comparison between *HER* and *HMR* is not significant. One possible explanation is that we do not have enough testing players, which is suggested by the fluctuation of the players' average ratings in the case of without DM (column w/o_DM in Table 1).

We allow the players to rate their narrative experience by whatever means they choose, instead of imposing a definition of enjoyment on the players. This adds strength to the results by showing robustness to individually differing beliefs about enjoyment. Although our personalized DM algorithm is studied in a simple testbed, it represents one of the most important fundamentals of drama management: guiding the players in a branching story graph. Our 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.

Conclusions

In this paper, we describe a new DM algorithm that aims to maximize the players' expected enjoyment. Our DM 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 guiding the player to the selected story experience in an interactive storytelling system. Compared to DMs in previous research, our personalized DM significantly increases the players' story experience ratings and guidance successful rate in a testbed built with CYOA stories.

Improving player experience is an important goal for the DM in interactive narrative. Although personalized drama management has not been well explored, we believe that building a personalized DM is essential to enhance the player experience. Our personalized DM can optimize each individual player's expected enjoyment while preserving his/her agency. Thus it is more capable of delivering enjoyable experience to the players in interactive narrative.

References

Bates, J. 1992. Virtual reality, art, and entertainment. *Presence: The Journal of Tele-operators and Virtual Environments* 1(1):133–138.

Bishop, C. M. 2006. *Pattern Recognition and Machine Learning*. Springer.

El-Nasr, M. S. 2007. Engagement, interaction, and drama creating an engaging interactive narrative using performance arts theories. *Interactions Studies* 8(2):209–240.

Lee, D. D., and Seung, H. S. 2001. Algorithms for non-negative matrix factorization. *Advances in Neural Information Processing Systems* 13:556–562.

Magerko, B., and Laird, J. E. 2005. Mediating the tension between plot and interaction. *Ann Arbor 1001* 48109–2110.

Mateas, M., and Stern, A. 2003. Integrating plot, character and natural language processing in the interactive drama Façade. In *Technologies for Interactive Digital Storytelling and Entertainment*.

Nelson, M. J., and Mateas, M. 2005. Search-based drama management in the interactive fiction Anchorhead. *Proceedings of the First AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment Conference* 99–104.

Peinado, F., and Gervas, P. 2004. Transferring game mastering laws to interactive digital storytelling. *Proceedings of the 2nd International Conference on Technologies for Interactive Digital Storytelling and Entertainment* 48–54.

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

Riedl, M. O., and Bulitko, V. 2013. Interactive narrative: An intelligent systems approach. *AI Magazine* 34(1).

Riedl, M. O.; Stern, A.; Dini, D. M.; and Alderman., J. M. 2008. Dynamic experience management in virtual worlds for entertainment, education, and training. *International Transactions on Systems Science and Applications*.

Roberts, D. L.; Nelson, M. J.; Isbell, C. L.; Mateas, M.; and Littman, M. L. 2006. Targeting specific distributions of trajectories in MDPs. *Proceedings of the Twenty-First National Conference on Artificial Intelligence*.

Thue, D.; Bulitko, V.; Spetch, M.; and Wasylishen, E. 2007. Interactive storytelling: A player modelling approach. *Proceedings of the third AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment Conference*.

Tipping, M. E., and Bishop, C. M. 1999. Probabilistic principal component analysis. *Journal of the Royal Statistical Society* B61(3):611–622.

Weyhrauch, P. W. 1997. Guiding interactive drama. *Ph.D. Dissertation* School of Computer Science, Carnegie Mellon University, Pittsburgh, PA. Technical Report CMU-CS-97-109.

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

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

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

Yu, H., and Riedl, M. O. 2014. Personalized interactive narratives via sequential recommendation of plot points. *IEEE Transactions on Computational Intelligence and Artificial Intelligence in Games* 6(2):174–187.

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