Data-Driven Personalized Drama Management

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

A drama manager is an omniscient background agent responsible for guiding players through the story space and delivering an enjoyable and coherent experience. Most previous drama managers only consider the designer's intent. We present a drama manager that uses data-driven techniques to model players and provides personalized guidance in the story space without removing player agency. In order to guide players' experiences, our drama manager manipulates the story space to maximize the probability of the players making choices intended by the drama manager. Our system is evaluated on an interactive storytelling game. Results show that our drama manager can significantly increase the likelihood of the drama manager's desired story continuation.

Introduction

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 (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. In many cases interactive narrative systems utilize 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). Given a number of things a player can do in a virtual world or computer game at any given time, the goal of a drama manager is to increase the likelihood that a player will experience an enjoyable and coherent narrative.

Prevailing approaches to drama management treat the DM as a surrogate for the human designer, acting to increase the likelihood that players will have narrative experiences that satisfy a set of criteria given by the game designer. This set of criteria provided by the human author is the only measures of quality for the player's interactive experience. Intuitively, players have different opinions on whether a narrative experience is enjoyable or not. In this work, we ask whether the DM can model players and use these models to increase the quality of individual players' narrative experiences. That is, we assert that a drama manager must *also* be a surrogate for the player by taking into account his or her preferences and behaviors.

Work on personalized DM has shown promise. Thue et al. (2007) show how a DM can choose narrative branches based on fixed player types. Sharma et al. (2010) use case based reasoning to choose narrative branches based on fixed feature vectors. Previously, we used a data-driven technique based on player ratings to learn players' preferences over trajectories through a branching story graph without fixed player types or fixed feature vectors (Yu and Riedl 2012). The strength of the data-driven approach is its ability to discover player types, thus making it applicable to a broader range of interactive narratives. However, in our prior work, the DM makes all branch choices, eliminating player agency, a key aspect of interactive narratives. In this paper, we build off our previous work with a drama manager that affords the player full agency while acting to influence the player such that he or she is more likely to make decisions that lead to an improved experience according to their own preferences.

Our Personalized Drama Manager (PDM) predicts players' personalized trajectories through a human-authored branching story graph, predicts players' choices, and manipulates the narrative to increase the likelihood that the player will make certain choices. The human designer's intention is expressed through the branching story graph while player agency is preserved because the player is never restricted from choosing particular branch. The paper presents the following novel contributions:

- An extension of the classical branching story graph to allow multiple options that lead to the same branch.
- A data-driven technique for predicting which option at each plot point an individual will choose.
- An algorithm for manipulating a branching story graph that increases desired player choices without restricting player agency.

We have evaluated our approach in a simplified testbed domain based on the original Choose-Your-Own-Adventure (CYOA) books. The human study results show that our drama manager can significantly increase the likelihood the players choose the intended story plot points.

Background and Related Work

Drama manager agents have been widely used to guide the users through an expected story experience set by designers. See Riedl and Bulitko (2013) for a recent overview of AI approaches to interactive narrative. Approaches to drama management include: search (Weyhrauch 1997; Nelson and Mateas 2005), planning (Riedl et al. 2008; Cavazza et al. 2009), and reactive behavior planning (Mateas and Stern 2003), case based reasoning (Sharma et al. 2010), and optimization (Roberts et al. 2006).

Most *personalized* DMs learn a model of play style using discrete play type categories. The PaSSAGE system (Thue et al. 2007) 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 game player types: Fighters, Power Gamers, Tacticians, Storytellers, and Method Actors. Peinado and Gervás (2004) use the same player types, while Seif El-Nasr (2007) uses heroism, violence, self-interestedness, and cowardice as dimensions. Sharma et. al, use case based reasoning based on a fixed set of hand-chosen features to choose branches in the story graph. Dimensional player models are good for games for which validated models exist, but may not extend to other types of interactive narratives.

PBCF (Yu and Riedl 2012) is a data-driven technique for learning players' preferences over narrative experiences, applying collaborative filtering (CF) to players' ratings of narrative experiences. CF algorithms attempt to detect patterns in users' ratings; CF algorithms discover latent user types that explain and predict user rating behavior. Because the recommendation of a plot point depends on the history of plot points previously visited by the player, PBCF extends standard CF algorithms to solve *sequential* recommendation problems. Unfortunately, in order to maximize player experience, PBCF chooses branches for the player, eliminating player agency. That is, PBCF is a story generator instead of a drama manager. We extend PBCF by restoring player agency; players are able to freely choose options after every plot point, and no branches are pruned.

Our PDM assumes that an interactive narrative experience can be represented as a *branching story graph*, a directed graph in which nodes represent *plot points* and arcs represent *options* the player can choose from. A branching story graph thus specifies which plot points are allowed to follow other plot points; it encodes human authorial intent. For the purposes of a DM agent, a branching story graph provides the set of successor plot points at any given time. While the representation is simple, many other drama management plot representations are reducible to the branching story graphs (Riedl and Young 2006; Weyhrauch 1997; Nelson and Mateas 2005; Roberts et al. 2006).

A Personalized Drama Manager

In this section, we describe our Personalized Drama Manager (PDM) which models players' storytelling preferences and also models players' behavioral choices. We hypothesize that players, when faced by a set of options afforded by the game (i.e., the arcs on a branching story graph), choose options based on a variety of local cues—those that sound most interesting, that agree with personal motivations, or that sound most likely to lead to favorable outcomes. Furthermore, we observe that certain trajectories through the branching story graph are more preferred than others, and that these preferences are individualistic (Thue et al. 2007; Yu and Riedl 2012). Therefore, it is possible for players' local choices of options to be in conflict with their global future interests.

In this paper, we aim to build a DM that is capable of influencing players' choices of options such that he or she is more likely to experience a highly rated narrative according to their individual preferences. Our PDM approach is summarized as follows. First, we extend the branching story graph representation to allow for multiple options that branch to the same successor plot point. Second, given the specialized branching story graph, the PDM predicts the best trajectory for a particular player through the graph. Third, the PDM uses CF to predict which option a player will choose after each plot point. Finally, when a discrepancy is detected between plot point predicted to be chosen and successor plot point predicted to maximize player experience, the PDM manipulates the branching story graph by showing some options and hiding other options such that the player is more likely to choose the personalized plot point. However, at no point is a successor plot point ever made unavailable.

Authoring Multiple Options

To give our drama manager the ability to manipulate the branching story graph without making successor plot points unreachable by the player, we extend the branching story graph representation such that multiple options are allowed to point to the same child plot point. Let a branching story graph be a directed graph $G = \langle V, E \rangle$ such that $v_i \in V$ is a plot point for $i = 1...|V|$, and $e_{i,j}^{k} \in E$ is the kth edge from v_i to v_j representing a player option available after plot point v_i. We use $e_{i,j}^{*}$ to denote the *existence* of at least one edge from v_i to v_j and indicating that v_j is an immediate successor of v_i. Figure 1 shows an example branching story graph with multiple options. The left side of the figure only shows successor relations, while the right side of the figure zooms in on one particular branch to show multiple options. As a shorthand, we use letters to indicate the successor relation and letters with superscripts to indicate distinct options.

Ideally, there are multiple options between all plot points and their immediate successors. The goal of the drama manager is to 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 child plot point. For example, suppose the drama manager predicts that a particular player's optimal narrative trajectory through Figure 1 is through plot point 11. Suppose the

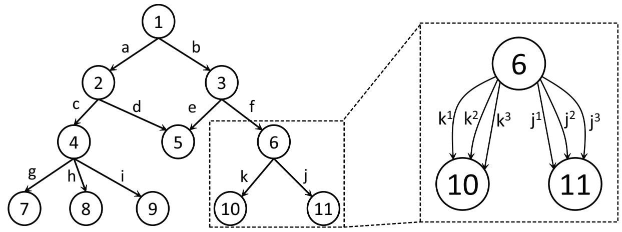

Figure 1: Example of a branching story graph with multiple options. Letters indicate successor relations between plot points, while letters with superscripts indicate distinct options the player may choose from.

drama manager further predicts that the player's preferences over options to be $k^1 > j^1 > k^2 > j^2 > k^3 > j^3$, such that the player is predicted to transition to plot point 10 instead. To intervene, a DM can present options j^1 and k^3 to the player, while suppressing the other options.

This simple extension to the conventional branching story graph gives a DM 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.

We believe that options should be authored to appeal to different motivations that they players might have, tapping into individual differences. In our own experiments, we have utilized the following motivational theories, drawn from Petty (1986) and Cialdini (2006):

- *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 we hypothesize that utilization of motivational categories will improve our drama manager's ability to learn players' preferences for options.

Player Option Preference Modeling

We assume that different players have different preferences over the options. For each player, if we know his/her preference for all the options in the extended branching story

Option	Player1	Player2	Player3	...
k^1	*	*	2	...
k^2	1	*	2	...
k^3	*	*	*	...
j^1	4	3	*	...
j^2	*	5	1	...
...	...	...	...	...

Figure 2: An illustration of the option-rating matrix. k^1, k^2, k^3, j^1, j^2, etc. represent the options in Figure 1. The stars represent missing ratings.

graph, it will be straightforward for the drama manager to select a subset of options to show. In this section, we describe how we train the drama manager to predict which options the player will prefer at any given plot point.

To predict players' option preference, we use collaborative filtering (CF) algorithms to build a players' option preference model. CF has been successfully applied in recommender systems to model user preference over movies, books, music, etc. (Su and Khoshgoftaar 2009). CF algorithms attempt to learn users' preference patterns from ratings feedback and predict new user's ratings from previous user's ratings which share similar preference patterns.

Applying CF algorithms to option preference, we have players rate the options presented after each plot point in the training phase. We then construct an option-rating matrix as in Figure 2. 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 option-rating matrix will be similar to the product rating matrix in traditional CF algorithms. The matrix will be sparse, containing a large number of missing ratings since we do not expect each player to read all the options in the extended branching story graph.

We investigated a variety of common CF training algorithms on the option-rating matrix, including: Non-negative Matrix Factorization (NMF) (Lee and Seung 2001; Zhang et al. 2006), probabilistic PCA (pPCA) (Tipping and Bishop 1999), K-Nearest Neighbor, 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 we have data for the player from another graph.

Drama Manager Algorithm

Our PDM attempts to influence players' trajectories through a branching story graph with multiple options per child plot point. We employ the insight that players will choose the options that sound the most interesting to them and that a DM can display options predicted to be more or less preferred by an individual to influence game play behavior. To achieve

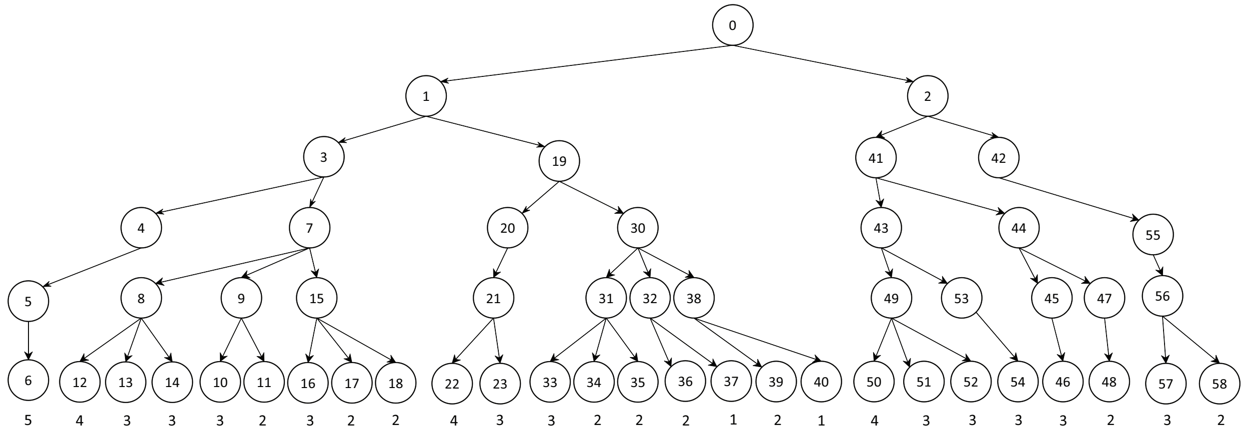

Figure 3: The branching story graph for the choose-your-own-adventure book: *The Abominable Snowman*. The digits at the bottom are the left-most score distribution we used in the evaluation.

this, a DM must have a model of players' preferences for options after each plot point to predict how players' will move through a branching story graph if left to their own device. This model is learned using one of the option-matrix CF algorithms described in the previous section.

Our PDM puts the model to use as follows. For each new player, our PDM must collect a few initial option ratings r. 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 and the player model is updated, the PDM uses the following algorithm to guide the player in the branching story graph. At each plot point in graph, the PDM performs the following steps.

1. Determine which child plot point the player should experience next.
2. Predict the player's preference for all options using the option preference model.
3. Display the highest rated option that points to desired successor plot point and the lowest rated option for each other successor plot point.
4. Player chooses an option.
5. Collect player's ratings for the displayed options. Include the ratings into r.
6. Display the corresponding child plot point according to the player's selection and go to step 1.

We assume that PBCF (Yu and Riedl 2012) or other player modeling algorithms can be applied in step 1. The details are beyond the scope of this paper. It is not strictly necessary to collect option ratings as in step 5. We do it in our system for the purpose of collecting as much data as possible to build more accurate player option preference models. With every new rating, the PDM will get better understanding of the current player's preference over the options.

Evaluation

To evaluate our PDM algorithm, we have conducted a study whereby our PDM attempts to influence players of an online choose-your-own-adventure interactive story. We hypothesize that our PDM algorithm will be able to significantly affect the behavior of players, as compared to a version of the interactive story with no drama management. We describe the Choose-Your-Own-Adventure story, online game environment, methodology, and results.

Stories and User Interface

We transcribed two Choose-Your-Own-Adventure books: *The Abominable Snowman* and *The Lost Jewels of Nabooti*, into two branching story graphs. The original stories were modified such that each possible narrative trajectory contains exactly six plot points. This was achieved by manually removing branches that led to "sudden death" outcomes and merging a few successive plot points. 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. Figure 3 shows the branching story graph of *The Abominable Snowman*.

We 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 earlier in the paper. In the final extended branching story graphs, there are thus three different options per successor plot point at every branching point except the lowest level, which we left unaltered. In total, there are 214 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 and all the options 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 4 shows our online interactive storytelling 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).

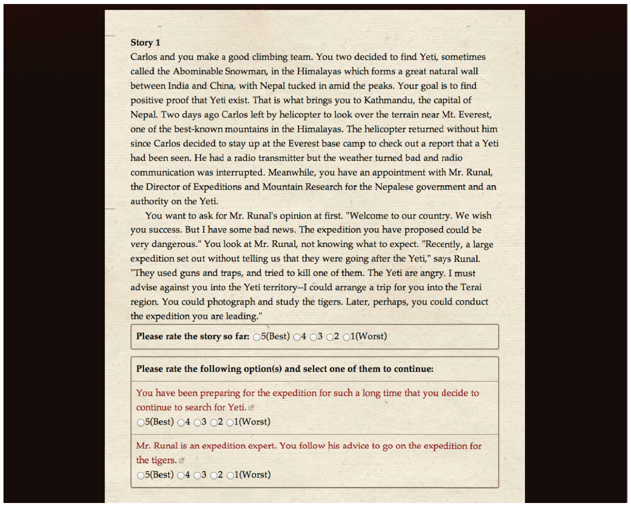

Figure 4: A screenshot of the interactive storytelling system.

Table 1: The average within-graph and cross-graph prediction accuracies of option selection for different algorithms.

Algorithm	within-graph	cross-graph
PPCA	0.7509	0.7589
NMF	**0.8123**	**0.8011**
KNN	0.7123	0.6931
KMeans	0.7080	0.6962

Training the Option Preference Model

We recruited 121 participants from Amazon's Mechanical Turk. Each player read 6 full-length stories. Each story was randomly started at the root of one of the two branching story graphs. Each story was presented plot-point by plot-point to the player as in Figure 4. At every branching plot point, the DM randomly picked one option for each successor plot point and the player was free to make a choice. We collected their ratings for all the options and stories they read. In total we had 121 valid players for a total of 726 play-throughs, and a 121×214 option-rating matrix.

Participants 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 randomly selected 80% of the training players and learned a player model with their rating data. For the remaining 20% of data, we computed their *within-graph* and *cross-graph* option selection prediction accuracies. To compute the within-graph accuracy, the DM built the initial rating vector for each player using the option ratings from the subtree with plot point 1 as its root in Figure 3. Then based on the player model and the initial rating vector, the PDM predicted the player's option selection behavior in the subtree with plot point 2 as its root. To compute the cross-graph accuracy, the PDM built the initial rating vector using the option ratings from the branching story graph of *The Lost Jewels of Nabooti*. Then the PDM predicted the players' option selection in the other branching story graph for *The Abominable Snowman*. We repeated the random split process 50 times and computed the average percent of time the PDM correctly predicted the players' selection. Table 1 shows the DM's within-graph and cross-graph prediction accuracies for different algorithms: pPCA, NMF (with 4 dimensions), K-Nearest Neighbor algorithm (with $k = 20$), K-mean algorithm (with $k = 4$). Our system is able to predict players' selection of options at greater than 80% accuracy.

Testing the Drama Manager

We recruited additional 72 participants from Mechanical Turk to evaluate our PDM's ability to guide the players in a full branching story graph. Each player read 6 full-length stories plot-point by plot-point.

For the first five stories, players explored the branching story graph of *The Lost Jewels of Nabooti*. This allowed us to incorporate some data about new players and update our model. As during model training, at every branching plot point, the PDM randomly picked one option for each successor plot point. During a player's sixth trial, the player played through the branching story graph of *The Abominable Snowman*. At each plot point of Figure 3, the PDM tried to guide the players to either the leftmost (43 participants) or the rightmost (29 participants) successor plot point by using the player model to choose two options pointing to the leftmost/rightmost successor plot point and choosing one option pointing to each remaining successor plot point.

We analyzed the effectiveness of our PDM in two different ways. First, we looked at the percentage of the time players choose options that corresponded with the PDM's desires. Second, we simulated the situation in which it targets certain nodes based on predicted scores from PBCF.

Frequency of Player Choices We looked at the percentage of the time the player chose an option at any plot point that leads to the leftmost/rightmost successor plot point. Table 2 shows the average guidance successful rate for different configurations. In our training data, we observe that players chose an option leading to the leftmost branch 53.91% of time and chose an option leading to the rightmost branch 44.52% of time. Note that some plot points have more than two successors and thus players may choose an option leading to neither the left nor right successor. When the PDM was configured to guide the players to the leftmost child plot point, it succeeded 74.07% of time ($p < 0.001$). When it was configured to guide the players to the rightmost child plot point, it succeeded 70.83% of time ($p < 0.001$).

Score Distribution Our second method for evaluating our PDM was to provide a score distribution over all leaf nodes in Figure 3. This simulates the situation where PBCF attempts to maximize players' experiences by predicting the ratings that players will give to leaves. This evaluation tells us how much more *utility* (in terms of player enjoyment) the presence of drama management will achieve over the lack of drama management. We constructed two score distributions, leftmost-score distribution and rightmost-score distribution,

Table 2: The average guidance successful rate for different PDM configurations.

Condition	Leftmost branch	Rightmost branch
No intervention	0.5391	0.4452
DM target: leftmost branch	0.7407	-
DM target: rightmost branch	-	0.7083
p-value	< 0.001	< 0.001

Table 3: The average scores of the full-length stories explored by the players for different PDM configurations.

Condition	Leftmost leaf	Rightmost leaf
No intervention	3.21	2.77
DM target: leftmost leaf	4.0	-
DM target: rightmost leaf	-	4.03
p-value	< 0.001	< 0.001

such that for any subtree in the graph, the leftmost or rightmost successor of the root is an ancestor of a higher-scoring leaf than any other successor of the root. The leftmost-score distribution is shown at the bottom of Figure 3. For this distribution, there is a leaf in the left subtree (e.g. leaf 6 in the subtree with root node 3) scoring higher than any leaves in the right subtree (e.g. the subtree with root node 19). The scores for the leaf nodes are in the range of 1 to 5.

Table 3 gives the average scores of the full-length stories explored by the players for different configurations. Without intervention, players followed trajectories resulting in an average score of 3.21 under the leftmost distribution, and an average score of 2.77 under the rightmost distribution. With the PDM attempting to guide the players to the leftmost branches, the system achieved an average score of 4.0 ($p < 0.001$). With the PDM attempting to guide the players to the rightmost branches, the system achieved an average score of 4.03 ($p < 0.001$).

Discussion

Our evaluation shows that our PDM can significantly influence the trajectories of players, regardless of how the target branch is chosen. When instructed to guide the player down certain branches (left or right), we find that the DM is able to significantly impact the likelihood that those branches are chosen. When instructed to guide the player to certain leaf nodes according to a score distribution, we find that the DM is able to significantly improve the utility of the player's experience according to the scoring metric. In future studies, we will use a PBCF-generated scoring distribution.

Player behavior is strongly affected by the story and the wording of options, often yielding a strong preference for a particular option at a particular plot point. For example, in our training data on *The Abominable Snowman*, we see that 84% of players prefer to transition from plot point 2 to plot point 41. Upon closer inspection, this branch involves the player continuing to hunt the Yeti versus abandoning the quest for a different objective; naturally people reading this particular book would have a preference for continuing the quest. Similarly, players preferred to transition from plot point 0 to plot point 2 75% of the time when left to their own devices. Our PDM will have a tougher time influencing players at plot points where players have a strong natural preference for one branch over another. If we exclude plot point 0 and 2 from testing analysis, the successful rate of guidance to the leftmost children will increase to 79.6%, and the successful rate of guidance to the rightmost children will increase to 75.86%.

Failure to guide the player at any given plot point reduces the optimality of the player's experience (according to some score distribution). In practice, if this happens our PDM attempts to guide the player to the next highest rated leaf node in the current subtree of the branching story graph. From the score comparison in the second evaluation technique, we show that our PDM is capable of guiding players to stories with higher simulated preference ratings.

Our evaluation shows that CF can be used for cross-graph prediction. Once our option preference model has been trained on two branching story graphs, it is possible for new players to provide a few ratings (as few as five stories worth of data) on a training graph and then receive accurate predictions on the other graph. Pragmatically, once the system is bootstrapped with training data from both graphs, new players only require a short familiarization phase before receiving personalized interactive narrative experiences.

Player agency is a critical aspect of interactive narrative. We did not ask players whether the drama manager reduced player perception of agency. However, since players are presented with options for all possible branches—no branches are outright denied to players—we believe that the appearance of player agency will be upheld.

Conclusions

Personalized drama management aims to deliver a personalized experience while preserving player agency. In this paper, we present a drama management system that can guide players in the branching story graph by manipulating the graph to increase the probability of desired story continuation without sacrificing player agency. In the future, our DM will operate in full conjunction with personalized story recommendation algorithms such as PBCF.

Player guidance and personalized drama management have not been widely explored. But we believe that they are essential parts of building a DM that is responsible for optimizing the player's experience in a game or virtual world. Our approach is capable of effectively influence players' choices while preserving the appearance of full player agency. A DM built in this way is more capable of bringing an enjoyable experience for the players.

Acknowledgments

We gratefully acknowledge the support of the U.S. Defense Advanced Research Projects Agency (DARPA).

References

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

Cavazza, M.; Pizzi, D.; Charles, F.; Vogt, T.; and Andre, E. 2009. Emotional input for character-based interactive storytelling. *Proceedings of the 8th International Joint Conference on Autonomous Agents and Multiagent Systems*.

Cialdini, R. B. 2006. *Influence: The Psychology of Persuasion (Collins Business Essentials)*. Harper Collins Publishers.

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.

Mateas, M., and Stern, A. 2003. 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*.

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

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*.

Petty, R. E., and Cacioppo, J. T. 1986. The elaboration likelihood model of persuasion. *Advances in experimental social psychology*.

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

Riedl, M., and Young, R. M. 2006. From linear story generation to branching story graphs. *IEEE Journal of Computer Graphics and Animation* 26(3):23–31.

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*.

Sharma, M.; Ontanon, S.; Mehta, M.; and Ram, A. 2010. Drama management and player modeling for interactive fiction games. *Computational Intelligence*.

Su, X., and Khoshgoftaar, T. M. 2009. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*.

Thue, D.; Bulitko, V.; Spetch, M.; and Wasylishen, E. 2007. Interactive storytelling: A player modelling approach. *Proceedings of the third 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*.

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*.