Achievement, Relationship, and Engagement (ARE): Using Deep Reinforcement Learning to Recover Users' Strategies and Motivations from Massive and Complex Online Game Behavioral Data

Leave Authors Anonymous
for Submission
City, Country
e-mail address

Leave Authors Anonymous
for Submission
City, Country
e-mail address

Leave Authors Anonymous
for Submission
City, Country
e-mail address

Leave Authors Anonymous
for Submission
City, Country
e-mail address

ABSTRACT
Massively multiplayer online role-playing game (MMORPG), such as World of Warcraft (WoW), attracts millions of users and many of them spend thousands of hours playing games online. Understanding their behaviors and the underlying motivations is of great interests to game designers and researchers, and also to parents and educators. We [applied] the deep reinforcement learning to model users' playing strategies as well as inverse reinforcement learning to model their motivations. We use a massive and complex online game behavioral dataset, World of Warcraft Avatar History (WoWAH), which recorded over 70,000 users' log data spanning 3 years time period. Our trained model not only can predict users' behaviors with a high level of accuracy. Moreover, it can also reveal users' motivation level changes in Achievement, Relationship, and Engagement.

INTRODUCTION
Understanding Massively multiplayer online role-playing game (MMORPG) games satisfaction mechanism and user behaviors could be non-trivial. As human players have a mixed feeling from different perceptions and they act not for a concrete, explicit objective such as winning an episode or taking high scores. While it's plausible to build a gaming bot from the player log data, to advance in the game, it ignores other dimensions of motivation which the players also care about. The game designers and researchers, also parents and educators, are, however, keen to reveal the underlying mechanism instead of just mastering the game.

Take World of Warcraft (WoW), which is one of the most successful MMO games in the world, as an example. It's massive, and multiplayer as millions of players subscribe the game platform, on which players can communicate with others, finish some quests together, compete with each other, and even build their own guilds. WoW also provides different kinds of races, careers and optional requests that could serve players extra flexibility to evolve different game strategies. With all those features the players receive many different dimensions of satisfaction which eventually compose the general reward system of the game.

For its huge research and commercial value to study game players' behavior, attentions has been drawn for both qualitative and quantitative studies. World of Warcraft Avatar History (WoWAH) dataset [7], published in 2011, records the location (in term of zone) of over 70,000 users every 10 minutes, together with some essential statistical analysis [1]. However, those analysis and the further studies based on the dataset are far from fully utilization of the data. While some of them use simple classifiers of clusterings for behavior analysis [15, 5], others focues on forcasting future events such as unsubscription of game or violation of terms [3, 16, 8]. Few of them are effectively modeling the interaction between the players and the game environment, neither on another dataset. On the other hand some studies tries to conduct studies on player motivations [4], but only able to apply some basic machine learning tools such as clustering.

Paste the appropriate copyright statement here. ACM now supports three different copyright statements:
• ACM copyright: ACM holds the copyright on the work. This is the historical approach.
• License: The author(s) retain copyright, but ACM receives an exclusive publication license.
• Open Access: The author(s) wish to pay for the work to be open access. The additional fee must be paid to ACM.
This text field is large enough to hold the appropriate release statement assuming it is single spaced.
Every submission will be assigned their own unique DOI string to be included here.

[1] we talk about the reason choosing this dataset in the first section of *Method*

Attribute	Value
Duration	1107 days
Sample Interval	Every 10 minutes
#Users	70,000+
Locations	One of 165 zones

Table 1. WoWAH Dataset Attributes

We re-employ the dataset in a totally different way: to model human-computer interaction from a reinforcement learning perspective. The human player and computer are characterized by a pair of agent and environment, as shown in Fig. ??. It gives an analogy between the traditional user experience model on the left, and the reinforcement learning scheme on the right. In the model, each player is modeled by an agent, whose zone transactions are regarded as the actions of the agent. Existing models in reinforcement learning succeed in modeling human behaviors by training the agent to maximize the explicitly defined reward. However, in WoW and most of the cases in CHI, the definition of reward is vague or partial, e.g. we can't tell if a play is optimal simply by looking at the experience gain credited to this play. Even if some could try building a reward function according to their knowledge of certain environments, it worth to note that figuring out the reward mechanism itself is a valuable task. In fact, figuring out the reward mechanism gives the most succinct description of the task. Especially for the developers of the environment, it's a way to give a quantitative representation of what they have designed.

The process of recovering the reward function is inverse reinforcement learning (IRL). In a data-driven manner, it recovers the underlying reward mechanism that induces the recorded user behaviors, simply by assuming that those players are trying to persue their rewards. Most of the IRL algorithms require we have access the dynamic of the environment, that is in WoW, we simulate the player's action and observe the feedback from the environment. Although we obviously don't have it, we address it by proposing an IRL algorithm based on purely off-policy training. The model analyzes the behaviors of the players, using the trajectories composed of player locations, and returns the reward function under which those trajectories are most plausible to happen. With the recovered reward, we train an agent to best mimic the general user behavior in WoWAH dataset. As our reward recovery algorithm works for any set of trajectories, we conduct future studies to character the dynamics of the game environment thus how design factors working on user behaviors, and compare different groups of users to examine the divergence of motivations.

RELATED WORK

Online Game Player Motivation And Rewards

One reason that online games appeal so many players is that it provides the environments for different kinds of play styles. The same online game may have respective meanings for different players. In the environment players gain certain kind of satisfactions (rewards) they want by conducting their actions. It's natural to assume a user's behaviors are highly correlated with the user's demanding satisfaction. And for game designers, finding the demanding satisfactions of users could make it easier to serve the users with better experience.

Components	Sub-components
Achievement	Advancement, Mechanics, Competition
Social	Socializing, Relationship, Teamwork
Immersion	Discovery, Role Playing, Customization, Escapism

Table 2. Components of game motivation

Previous study categories the satisfactions into three major aspects, namely, achievement satisfactions, social satisfactions, and immersion satisfactions. It was then divided into ten subcategories, briefed in Table. 2. More detailes are introduced in [18] and [17]. The weights associated with these ten kinds of satisfactions basically chartered a user's profile, and it would be very meaningful the weights can be recovered in a purely data-driven process, based on the user's playing history.

Player Profile Study

Reinforcement Learning and Apprenticeship learning

Reinforcement learning (RL), especially deep reinforcement learning (DRL), is an [] domain inspired by behaviorist psychology. In RL, the agent performs in an online environment (in which the agent conducts a sequence of actions and get a reward for each of its actions) and evolves itself from the pains and pleasures it received from its previous actions. In the environment the agent has to deal with the tradeoff between exploration and exploitation, which presents in many real life situations. For example, to level up quickly in a game in a long-term perspective, the player has to conduct some preparation work which does not benefit its leveling up in the near future.

RL is able formulate many online problems to model users' behavior. For example, the trained agent conducts superior play compared with professional human in the game of Go [14], Atari 2600 [10], and Poker [6]. Also used to model users' clickings and browsings for online shopping websites, and how users behaves in many optimal control problems etc. (Note: I want an example more related to wowah). It's worth to note that recent advancement of DRL makes it possible to handle complex environment with high-dimension observation and state spaces, making it possible to model a wide variety of real-life problems.

In most cases of user behavior analysis and modeling, the reward scheme of the environment is not explicitly given. For example, the user experience of a software or a online game is the combination of different kinds of satisfication. Instead, apprenticeship learning [12, 1, 11], also known as apprenticeship via inverse reinforcement learning (AIRP), tries to recover the underlying reward function using the observed behavior of the agent, e.g. the trajectories of players in a online game. Applying AIRP on users' behavior log could help the developers understand the intention of the user, the divergence between different group of users, and the dynamic of user profiles.

METHOD

Dataset Description

To model the user behavior, we treat user as the agent who conduct an action every 10 minutes; during the 10-minute interval, the user decide the zone to stay in the next interval. If the user has been in multiple zones in a single interval, only the zone with longest stay duration was recorded. The actions are represented by zone IDs, ranging from 0 to 164, corresponding to 165 zones existing during Jan 2006 to Jan 2009 in WoW. When counting the index of interval, we ignore those minutes that the user's offline.

We use World of Warcraft Avatar History (WoWAH) dataset [7], a dataset collected from realm *TW-Light's Hope* during 1st Jan 2006 and 10th Jan 2009, containing 70,055 users after we filtering out those with too short playing history. Each user has spent 440.4 time intervals online on average. The dataset contains many different kinds of users: both novice and expert, guild members and isolated players, low-level and high-level players etc., with their respect class, and race in game. The detailed attributes are listed in Table reftbl:attributes.

During the gameplay the user is able to observe its current game states, including all the attributes recorded in the dataset. We construct the observation vector of the agent using the sequence of attributes extracted from its game plays in the most recent session (since the lastest log on). Instead of using the raw, concatenated vector of those attributes, we employ a preprocess to reduce the redundency, and make those decisive information more explicitly shown to the agent.

To model the user's behavior in a reinforcement learning perspective, we define the reward function which the agent tries to optimize during the gameplay. The reward is separated into five parts $r_1, \cdots r_5$, corresponding to five different kind of satisfactions defined in [18]. The construction of $r_1, \cdots r_5$ are illustrated in Table 3. The value of different kinds of satisfactions are normalized into the same scale.

Reward Recovery

We apply apprenticeship via inverse reinforcement learning, in order to recover the underlying reward mechanism of the players, using the trajectory τ of a user (or a group of users). Assume the total reward a user receives is the convex combination of the five rewards listed in Table. 3. Let $f_t = (f_t^1, f_t^2, f_t^3, f_t^4, f_t^5)$ be the rewards the user received at time t, we have the total reward the user receives at time t

$$r_t = \phi^T f_t,$$

where ϕ is the combination weight with $||\phi||_1 = 1,\ \phi \geq 0$. Assume at each time step, the user tries to take an action a according to the current game state so as to maximize the best expected cumulative discounted (with discount γ over time) reward (known as the action-value function) $Q^*(s, a)$, where

$$Q^*(s, a) = \mathbf{E}[R_t | s_t = s, a_t = a | \pi^*]$$

and

$$R_t = \sum_{t' \geq t} \gamma^{t'-t} r_{t'} = \sum_{t' \geq t} \gamma^{t'-t} \phi^T f_{t'}.$$

The term π^* indicates optimal policy, described by a distribution $\mathbb{P}(a|s)$ over the feasible action space $\mathcal{A}(s)$. In this setting, the weight ϕ must satisfy that the action the user has taken must induce a larger Q^* value than any other valid action would have done. This optimality infers that

$$Q^*(s, a) \geq \max_{a' \in A(s)} Q^*(s, a') \tag{1}$$

is satisfied for all (s, a) pairs appeared in the user's trajectory. Consider the existence of possible sub-optimal actions conducted by the user, we introduce slack variables $\xi_{s,a}$ into the problem formulation. Let $\xi_{s,a}$ be the difference of the actual action-value $Q^*(s, a)$, and the largest possible action-value $\max_{a' \in A(s)} Q^*(s, a')$ whenever Eq. (1) is not satisfied, and zero otherwise. We minimize the summation of $\xi_{s,a}$ over the recorded user's

Sat.	**Category** & Definition
f^1	**Advancement** The speed the user collecting experience and leveling up in game. It's the most common reward a user could receive
f^2	**Competition** The satisfaction the user get by joining battleground or arena and competing with human opponents
f^3	**Relationship** The long-term relationship with the user's guild, which is quantified by the time elapse since the user join its guild
f^4	**Teamwork** The satisfaction the user get by playing in a zone which is featured by teamwork, e.g. Battleground, Arena, Dungeon, Raid, or a zone controlled by The Alliance.
f^5	**Escapism** Escapism begins to cumulate if the user has been online for 4 hours or has been regularly login to the game for 20 days.

Table 3. Different types of satisfactions

trajectory

$$-C\sum_{s,a}\left[\min(0, Q^*(s,a) - \max_{a'\in A(s)\setminus a} Q^*(s,a'))\right], \quad (2)$$

which is then formulated into the following linear programming (LP) problem

$$\begin{aligned}
\underset{\phi,\xi}{\text{minimize}} \quad & C\sum \xi_{s,a} \\
\text{subject to} \quad & \phi^T(Q(s,a)-Q(s,a')) \geq -\xi_{s,a},\ \forall (s,a)\in\tau, a'\in\mathcal{A}(s)\setminus a \\
& \phi \geq 0,\ ||\phi||_1 \geq 1 \\
& \xi_{s,a} \geq 0\ \forall (s,a)\in\tau.
\end{aligned} \quad (3)$$

In Eq. (3), $Q(s,a) = (Q^1(s,a), Q^2(s,a), Q^3(s,a), Q^4(s,a), Q^5(s,a))^T$, and

$$Q^i(s,a) = \mathbf{E}[\sum_{t'\geq t}\gamma^{t'-t} r^i_{t'}|s_t = s, a_t = a|\pi^{i,*}] \quad (4)$$

is the action-value function, when the user only takes the r^i into account and ignores all four other kinds of satisfactions. The LP formulated above is equavalent to minimizing Eq. (2) because by definition we have $Q^*(s,a) = \phi^T Q(s,a)$

Action-Value Function Approximation

To solve LP (3) it suffices to estimate $Q^i(s,a)$. As the number of feasible states s could be arbitrary large, for an user in WoW, it's impossible to enumerate over the state space. Instead, we use deep-Q networks (DQN) [10] to approximate the Q^i functions. The neural network takes s as input, and output the Q value for every action a. We use the same network architecture for $i = 1,\ldots,5$, illustrated in Fig. **??**. The categorical elements in s are firstly processed by an embedding layer [9], while the numeral elements are fed into an fully connected (FC) layer with rectifier non-linearity. The output of embedding layer and FC layer are then concatenated and fed into another FC layer with rectifier nonlinearity. A final FC layer is applied to compute the $Q(s,a)$ value for each action a.

Denote the trainable parameters in the Q-network as θ^i, we optimize over θ^i in order to approximate Eq. (4). The key observation of Q-learning is that, the action-value function, by its definition, should satisfy the Bellman equation. That is, if the user takes action a and the state turns into s_{t+1} from s_t, we have

$$Q^i(s_t,a_t) = r_t + \gamma\max_{a'} Q^i(s_{t+1},a'). \quad (5)$$

Q-learning tries to find the action-value function satisfying Eq. (5), by minimizing the squared difference L between both sides of the equation. Let

$$L^i = \mathbb{E}_{s_t,a_t,r_t,s_{t+1}}\frac{1}{2}(Q^i(s_t,a_t) - r_t - \gamma\max_{a'} Q(s_{t+1},a'))^2.$$

Since L^i is differentiable with respect to θ^i, θ^i can be updated via stochastic gradient descent, by

$$\theta^i \leftarrow \theta^i - \alpha\frac{\partial L}{\partial\theta^i}\Big|_{s_t,a_t,r_t,s_{t+1}}$$

We take advantage of the algorithm introduced in [], that the target network only get updated periodically, which is important for the stability of DQN training. The derivative of L with respective θ^i becomes

$$\frac{\partial L}{\partial\theta^i} = \mathbb{E}\left[(Q^i(s_t,a_t) - r_t + \gamma\max_{a'} Q(s_{t+1},a'|\theta^{i-}))\cdot\frac{\partial Q^i(s_t,a_t)}{\partial\theta^i}\right]$$

where θ^{i-} is the network parameter which is assigned current θ^i value periodically during training.

Agreement Between Q^1 and Advancing Behaviors

We conduct a quantitative evaluation of our learned Q^1 network. Consider collecting experience and getting level up is one of the major objective for players, we evaluate if the actions predicted by Q^1 agree with the moves conducted by those advancing players (who level up quickly). At state s, the predicted action is the one with the largest action-value $Q^1(s,a)$, i.e.,

$$a = \text{argmax}_{a'} Q^1(s,a').$$

For the (s,a) pairs extracted from the top 200 (in leveling up speed) players' trajectories, the prediction accuracy is 0.49 with the total number of feasible actions $|\cup_s\mathcal{A}(s)| = 156$, corresponding to 156 different zones in WoW existed during Jan 2006 - Jan 2009. It's competitive with 0.53, using policy cloning [2, 13] with classifier, and overperforms 0.23 if we approximate the Q function using linear function. The quality of Q^1 is decent.

RESULTS

Different from previous approaches on Atari games [10] and zero-sum games [14, 6], our approach models the user behavior instead of simply persuing advancement. Using Table. 3 and solve LP (3) on a dataset randomly drawn from the whole WoWAH dataset, the universal underlying reward mechanism of the player community is revealed as $\phi = (0.61, 0.16, 0.07, 0.05, 0.10)^T$. In another words, when conducting a behavior, in general the user's consideration is composed of 61% there advancement, 16% their competition, 7% their relationship, 5% their teamwork, and 10% their escapism. The results is illustrated as a spider map in Fig. **??**.

Armed with the model we conduct three different kinds further studies on WoWAH dataset. First we show that our model better describes the user behavior, than those who takes only single satisfaction metric. After recovering the underlying reward mechanism of the environment, our model could predict the move of the users, instead of just suggesting the best move for advancing (as Q^1 does). We then show the divergence of the underlying reward machenism between different groups of people, which is the causality driving the divergence of user behaviors. For example as the level of a player becomes higher, it tends to cares more about relationship than advancement. Finally we show, using our model, the evolution of underlying user satisfaction over time. Especially, observing the dynamics around the release of patch *Fall of the Lich King*[2], it shed some light on quantitative models of the game design.

Predicting the Users' Behavior

game

Armed with the reward machenism we are able to model users' behavior instead of creating an agent to play the game itself. Consider that the users' motivation of gameplay is more than leveling up,

Another interesting observation is the tradeoff between exploitation and exploration. In fact, many users' actions are sub-optimal, in the sense that they care too much about the satisfaction in the near future.

Behavior Divergence between Different Groups of User

Behavior Evolution over Time

The way the user

[2] http://wowwiki.wikia.com/wiki/Patch_3.3.0

REFERENCES

1. Pieter Abbeel and Andrew Y Ng. 2004. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning.* ACM, 1.
2. R Amit and Maja J Mataric. 2002. Parametric primitives for motor representation and control. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, Vol. 1. IEEE, 863–868.
3. Christian Bauckhage, Anders Drachen, and Rafet Sifa. 2015. Clustering game behavior data. *IEEE Transactions on Computational Intelligence and AI in Games* 7, 3 (2015), 266–278.
4. Jonathan Bell, Swapneel Sheth, and Gail Kaiser. 2013. A large-scale, longitudinal study of user profiles in world of warcraft. In *Proceedings of the 22nd international conference on World Wide Web companion.* International World Wide Web Conferences Steering Committee, 1175–1184.
5. Anders Drachen, Christian Thurau, Rafet Sifa, and Christian Bauckhage. 2014. A comparison of methods for player clustering via behavioral telemetry. *arXiv preprint arXiv:1407.3950* (2014).
6. Johannes Heinrich and David Silver. 2016. Deep Reinforcement Learning from Self-Play in Imperfect-Information Games. *arXiv preprint arXiv:1603.01121* (2016).
7. Yeng-Ting Lee, Kuan-Ta Chen, Yun-Maw Cheng, and Chin-Laung Lei. 2011. World of Warcraft avatar history dataset. In *Proceedings of the second annual ACM conference on Multimedia systems.* ACM, 123–128.
8. Jing-Kai Lou, Kuan-Ta Chen, Hwai-Jung Hsu, and Chin-Laung Lei. 2012. Forecasting online game addictiveness. In *Proceedings of the 11th Annual Workshop on Network and Systems Support for Games.* IEEE Press, 6.
9. T Mikolov and J Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems* (2013).
10. Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, and others. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529–533.
11. Andrew Y Ng, Stuart J Russell, and others. 2000. Algorithms for inverse reinforcement learning.. In *Icml.* 663–670.
12. Nathan D Ratliff, J Andrew Bagnell, and Martin A Zinkevich. 2006. Maximum margin planning. In *Proceedings of the 23rd international conference on Machine learning.* ACM, 729–736.
13. Claude Sammut, Scott Hurst, Dana Kedzier, Donald Michie, and others. 1992. Learning to fly. In *Proceedings of the ninth international workshop on Machine learning.* 385–393.
14. David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, and others. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature* 529, 7587 (2016), 484–489.
15. Mirko Suznjevic, Ivana Stupar, and Maja Matijasevic. 2011. MMORPG player behavior model based on player action categories. In *Proceedings of the 10th Annual Workshop on Network and Systems Support for Games.* IEEE Press, 6.
16. Ruck Thawonmas, Keisuke Yoshida, Jing-Kai Lou, and Kuan-Ta Chen. 2011. Analysis of revisitations in online games. *Entertainment Computing* 2, 4 (2011), 215–221.
17. Nick Yee. 2006a. The demographics, motivations, and derived experiences of users of massively multi-user online graphical environments. *Presence: Teleoperators and virtual environments* 15, 3 (2006), 309–329.

18. Nick Yee. 2006b. Motivations for play in online games. *CyberPsychology & behavior* 9, 6 (2006), 772–775.