Proceedings of the Seventeenth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment
(AIIDE 2021)

Generating Tailored Advice in Video Games through Play-Style Identification and Player Modelling

Branden Ingram
School of Computer Science and Applied Mathematics
University of the Witwatersrand, Johannesburg, South Africa
branden.ingram@wits.ac.za

Abstract

Recent advances in fields such as reinforcement learning have enabled the development of systems that are able to achieve super-human performance on a number of domains, specifically in complex games such as Go and StarCraft. Based on these successes, it is reasonable to ask if these learned behaviours could be utilised to improve the performance of humans on the same tasks. However, the types of models used in these systems are typically not easily understandable, and can not be directly used to improve the performance of a human. My research looks to address these difficulties by developing a system that can provide advice tailored to a player's style in a video game setting. This system would be particularly useful for improving tutorial systems that quickly become redundant lacking any personalization. I have already developed an unsupervised approach to identifying different play-styles present in a video game. I look to use this knowledge to train agents which can demonstrate optimal behaviour for each style. It is hoped that this information can be utilised to generate useful advice based on the differences between the current performance of a player and the corresponding expert agent in their style.

Introduction

It is simple to define learning as a process of acquiring new understanding, knowledge, behaviours, skills, values, attitudes, and preferences. However, determining how to help someone through this process is a much harder problem. This is partly due to a task's inherent difficulty but also due to the variety of styles with which a task can be approached. For example, video games such as StarCraft or DotA are complex domains where the path to improvement for an average user is not obvious. Additionally, a player may have a preference for exploring the minutiae of a game, or for completing it as quickly as possible. This makes the problem of tutoring more challenging as it is not guaranteed that one solution will benefit all people.

The diversity of possible play-styles can be significant, meaning that different players could play in very different ways. My goal is to develop a system to generate useful advice tailored to the different characteristics of the individual user. To achieve this goal the system should be able to learn to identify the varying play-styles present in a domain.

There are multiple benefits in identifying the play-style of an individual player as they engage in a game. From the player's perspective, knowing their style could be used to assist in the real-time tailoring of game mechanics for their needs and preferences. In the case of the designers, they gain an insight into whether or not their players are using the features and playing the game in an intended manner. There is an added benefit of being able to tailor tutorial mechanics and in-game tips to the type of player identified.

However, since my primary goal is to provide relevant advice to players, it is not enough to simply know how an individual will approach a problem. A model which describes the optimal solution is still required. Developments in Reinforcement Learning (RL) have led to being able to learn optimal solutions in many complex domains including video games. I look to leverage these RL algorithms to train our optimal agents. It will be these optimal agents that I utilise to determine the desired behaviour we wish players to achieve.

Generating understandable advice would be useful in many domains such as education or, specific to this research, gaming. Additionally, tutorial systems built within video games have traditionally been handcrafted and generic. These systems quickly become useless and lack any form of adaptation to different users who each have their way of playing as they each find enjoyment from different aspects of the game. This research proposes a system that can identify possible strategies from game-play trajectories of many different players. These strategies will be used in conjunction with a learned model of a particular user to generate beneficial tailored advice for that user. As a result, my approach will be able to handle different users as well as be able to adapt to the changing ability and strategies of the user.

Related Work

Play-Style Identification

Traditionally, play-style identification has been approached through player modelling, which is the study of computational models of players in games. This includes the detection, modelling, prediction and expression of human player characteristics which are manifested through cognitive, affective and behavioural patterns (Yannakakis et al. 2013). In model-based approaches, a player model is built on a theoretical framework where a preexisting understanding of

the domain is leveraged. For example, Anderson, Boyle, and Reiser (1985) developed a tutoring system based on a cognitive framework of learning. I look to achieve similar outcomes in an unsupervised fashion without the need for leveraging preexisting domain knowledge. This is similar to a model-free approach where a mapping (model) between (player) input and a player state representation is learned (Yannakakis et al. 2013). To achieve this observations are collected and analysed to generate a model without strong initial assumptions of its structure. Drachen, Canossa, and Yannakakis (2009) trained emergent self-organising maps to classify players into four styles. These techniques have been effective, however, have utilised meta-data summaries over raw user data experiences. I looked to achieve the same goal by analysing raw trajectory data without the need for extensive feature engineering.

This analysis is based on trajectory clustering, which is the comparing and grouping of trajectories based on a measurement of similarity. Techniques such as k-Means and hierarchical clustering have been utilised to perform the clustering step (Nanni 2002). However, these techniques are limited to the original data space and tend to be ineffective when input dimensions are high (Steinbach, Ertöz, and Kumar 2004). Additionally, Neural Network approaches (ART (Carpenter and Grossberg 1987)) or self-organising maps (SOM (Kohonen 1990)) have been utilised. Wang et al. (2004) used SOMs to cluster time-series features, however, SOMs do not handle trajectories of unequal length, as the dimensions of the weight vectors are fixed (Liao 2005). In addition to these, deep learning techniques have to been applied to unsupervised clustering (Xie, Girshick, and Farhadi 2016). Xie, Girshick, and Farhadi (2016) used an autoencoder which compressed data making clustering more feasible. I looked to utilise a similar approach to handling video game trajectories which have the added characteristic of being variable length with multi-dimensional time steps.

Game Modelling

Identifying play-style is only half the problem as an expert machine trained to be the best variation of that style is also required. Creating expert machines in video games has commonly been conducted using Reinforcement Learning. The success of this approach was demonstrated by TD-gammon which achieved superhuman levels through self-play (Tesauro 1995). Riedmiller (2005) would later use a Q-learning based Neural Network, such as Neural-fitted Q iteration (NFQ) which used the RPROP algorithm to update the Q-network. However, this update incurred an operational cost that was dependant on the size of the dataset. To avoid this, Mnih et al. (2013) used stochastic gradient descent resulting in low-cost updates and allowing their DQN to be applied to a larger dataset. These methods require the definition of a rich reward function that is able to accurately describe the dynamics of expert behaviour. Reward functions have been learned through a process of Inverse Reinforcement Learning (IRL) (Abbeel and Ng 2004). These approaches will be utilised to develop my play-style centric models.

Past Work

My previous work has focused on learning how to identify the different styles present in trajectory data through a deep unsupervised clustering approach. These trajectories being game state sequences collected from multiple play-throughs.

Play-Style Identification

This approach is based on two key steps. First, I utilise an autoencoder network to project a trajectory into a lower-dimensional latent representation. I can then perform clustering on this latent space to discover clusters corresponding to related trajectories in this space. The generalised system is depicted in Figure 1.

Trajectory Encoding: An autoencoder works to reconstruct each original input trajectory ($x_i \in X$) after first encoding it as a lower-dimensional state ($z_i \in Z$). My specific model is a temporal autoencoder containing LSTM layers. This allows us to be able to process trajectories of varying length by feeding the state at each time step into its own LSTM cell. These cells, pass on the important information in sequence until finally outputting a fixed-sized vector representative of our latent space (Z). The network is trained using back-propagation through time (Hochreiter and Schmidhuber 1997).

Trajectory Clustering: Having projected the trajectories into the latent space, I then cluster them. This clustering step is performed on a set of (x, z) pairs generated $\forall x \in X$ and z. (x, z) is clustered with respect to z to form predicted labels y'. Since z_i is a representation of x_i I can use y'_i as the cluster label for the original data. Clustering using Z enables the use of most clustering algorithms as there is no longer an issue of varying length or temporal features.

This approach to play-style identification was tested on data generated from multiple domains. Here, the accuracy of predicting the correct play-style was measured for both complete and partial trajectories. The results demonstrated that the generated clusters matched the ground-truth underlying play-styles. This is visualised in Figure 2 where heatmaps of the original trajectories separated by play-style are compared to the clustered play-styles. I also demonstrated the model's ability to recover the play-style from partial trajectories, demonstrating the usefulness of the system during gameplay. Not only will this system be used to identify the style of active users but it is also a requirement when training agents who act as experts for a given play-style.

Future Work

My future work focuses on developing each of three remaining components depicted in Figure 1 namely; The Game Modeller, Player Modeller and Advisor.

Game Modeller

The game modeller as depicted in Figure 1 is a system that looks to generate a set of optimal policies $(\pi_i^*(S))$ for each play-style (i). I approximate $(\pi_i^*(S))$ through iteratively updating our valuation of states (S) based on a reward function (R) (Mnih et al. 2013). In order to train a set of plays-

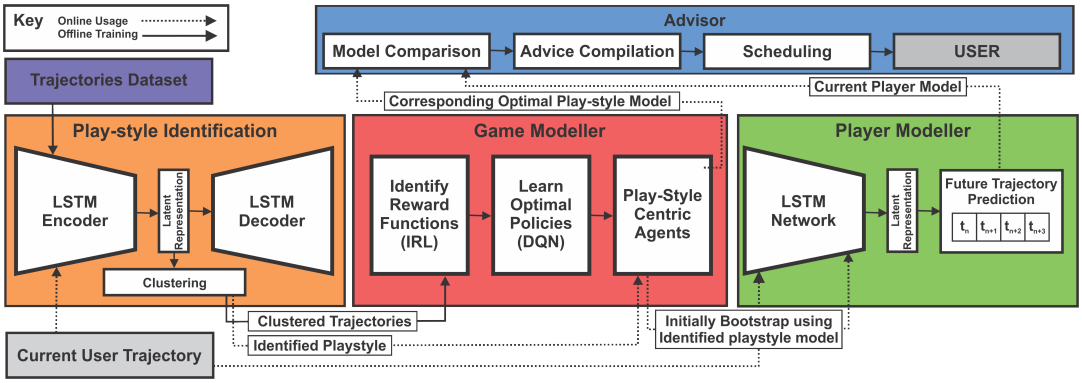

Figure 1: Overall model for tailored advice generation indicating offline play-style identification and expert model training from game-play dataset as well as online player modelling and advice generation from user trajectories.

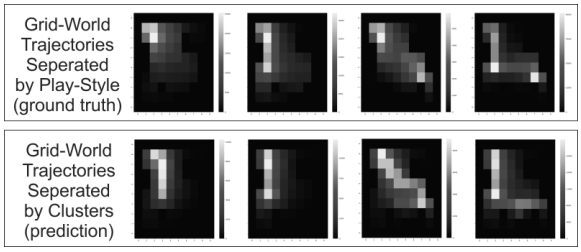

Figure 2: Visualisation of clustered trajectories (bottom) corresponding to the original trajectories (top).

tyle centric agents, I require a set of reward functions (R_i) defined for each play-style. I will initially handcraft these reward functions by rewarding states unique to a particular play-style identified through the identification system. Abbeel and Ng (2004) demonstrates the ability to recover reward functions from sampled expert trajectories. This is in essence a supervised approach to learning the motivations present in data. I also look to employ IRL in order to learn the different reward functions. However, by training on the clustered trajectories from the play-style identification system we can learn the reward functions in an unsupervised manner.

Player Modeller

An understanding of how the user plays is required to generate tailored advice. To accomplish this, I will train an LSTM network to be able to predict the future states from a partial trajectory of an individual. Ordinarily, this would require a significant amount of data from a user to train an accurate model of that user. Therefore, advice would only be beneficial once enough data was collected. This lag will be minimised by bootstrapping this learning process by leveraging my trained play-style identification and game modeller systems. By doing so I can identify the user's play-style (k) from partial trajectories and then use the corresponding $(\pi^*_{i=k}(S))$ as the base model of the player.

Advisor

The last piece of work will be the advisor system. This will utilise the information generated by the other three components and process it into a form that the user can use to improve their effectiveness. This processing involves the comparing of the generated player model with that of the associated expert play-style centric model. The advice would then be to simply encourage the user to adjust their play-style according to the following items:

- The user's relative skill level within their current play-style. This will be determined by calculating a measure of distance between a users current trajectory and a trajectory generated by their corresponding expert.
- The characteristics that differ between their model and the corresponding play-style centric model. These characteristics represent the states which differed the most from that of their corresponding expert trajectory.

Once the advice has been generated it will be sent to the user in the form of a text notification. The best time to intervene will be determined through the use of experimentation. Further studies could look into learning the appropriate time and form in which advice should be delivered.

Conclusion

My research proposes an end-to-end method to reduce the difficulty user's experience when trying to improve in video games. To this end, I look to develop a system that learns a model of the user by leveraging their identified style. It is hoped that useful information can be garnered through comparison between this user model and a pre-trained expert model which corresponds to the identified play-style. The ultimate goal being to be able to utilise this information to create high-level tailored advice which can help the user improve their performance. Although I look to experiment in video game domains, it is hoped that this approach can be applied to any domain where the path from novice to expert is different and unknown for each individual.

References

Abbeel, P.; and Ng, A. Y. 2004. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, 1.

Anderson, J. R.; Boyle, C. F.; and Reiser, B. J. 1985. Intelligent tutoring systems. *Science* 228(4698): 456–462.

Carpenter, G. A.; and Grossberg, S. 1987. A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer vision, graphics, and image processing* 37(1): 54–115.

Drachen, A.; Canossa, A.; and Yannakakis, G. N. 2009. Player modeling using self-organization in Tomb Raider: Underworld. In *2009 IEEE symposium on computational intelligence and games*, 1–8. IEEE.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8): 1735–1780.

Kohonen, T. 1990. The self-organizing map. *Proceedings of the IEEE* 78(9): 1464–1480.

Liao, T. W. 2005. Clustering of time series data—a survey. *Pattern recognition* 38(11): 1857–1874.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; and Riedmiller, M. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* .

Nanni, M. 2002. *Clustering Methods for Spatio-temporal Data: Ph. D. Thesis*. CS Dept., Univ. of Pisa.

Riedmiller, M. 2005. Neural fitted Q iteration–first experiences with a data efficient neural reinforcement learning method. In *European Conference on Machine Learning*, 317–328. Springer.

Steinbach, M.; Ertöz, L.; and Kumar, V. 2004. The challenges of clustering high dimensional data. In *New directions in statistical physics*, 273–309. Springer.

Tesauro, G. 1995. Temporal difference learning and TD-Gammon. *Communications of the ACM* 38(3): 58–68.

Wang, X.; Smith, K. A.; Hyndman, R.; and Alahakoon, D. 2004. A scalable method for time series clustering. *Unrefereed research papers* 1.

Xie, J.; Girshick, R.; and Farhadi, A. 2016. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, 478–487. PMLR.

Yannakakis, G. N.; Spronck, P.; Loiacono, D.; and André, E. 2013. *Player modeling*. Dagstuhl Publishing.