Assignment 6 - Project Proposal: Improving meIRL-based motion modelling in video games using general player classification

Inge Becht

Supervised by Sander Bakkes

April 26, 2013

1 Literature Review

One of the challenges in game AI research is how to create artificial agents that can make decisions based on a game situation, without the human players having the feeling that the AI is cheating. Creating such a fair AI can be achieved by giving it imperfect knowledge of the game world, for example by keeping parts of the game map unknown or by limiting the knowledge of the whereabouts of the opponent. In the latter case, the AI would benefit from creating motion models that are able to predict the position of an opponent and to reason with this uncertain knowledge.

The creation of such motion models that deal with position uncertainty has been the topic of various researchers. In the works of [1] two different models for opponent position prediction are tested. The idea of the research is to see how well the predictions can be made, as well as to test how human like the predictions made are. They test this for both Hidden Semi Markov Models and Particle filters, by letting people predict opponent position in the game Counter Strike, and then look how well these models perform in regard to human prediction. In [3] the research is more directed towards how much integrating a particle filter in an AI system enhances performance of an AI. The experiments were conducted on a Real Time Strategy game, and gave promising results, implying that predictional reasoner can improve the performance of an AI. In the paper [2] both the combination of making predictive models for opponent positions is discussed, as well as a way of intercepting the opponents in a game by using this information. The predictive models are made using a particle filter both with Maximum Entropy Inverse Reinforcement Learning (meIRL) and Brownian motion to test which works best. Not surprisingly, meIRL seems to give the most accurate result and helps to intercept the opponent the best. For interception 3 different heuristics are tested.

2 Research Question

In my thesis I want to improve upon the last presented technique of motion modelling, by answering the question

> To what extent can player classification improve meIRL-based motion modelling in video-games?

To answer this question, not only is there a need for an implementation of the motion model itself, there is also a need for building a reasoner that takes actions on the prediction of the motion model, as well as a classifier that classifies the behaviour. These three elements will also play a part in the evaluation, as will be explained later.

3 Method and Approach

Here a short overview of the motion model, classifier and reasoner are given, but first some mention of the game domain and the tools used.

3.1 Capture the Flag

Capture The Flag (CTF) is a well known combat objective found in the First Person Shooter video game genre, and will be the domain in which the experiments are conducted.

In short, two enemy teams try to catch the other team's flag from its spawn area and bringing it back to base. Both teams can shoot at each other and in case the flag bearer gets killed, the flag will stay at its position until its team retrieves it or another enemy opponent catches it and becomes the new flag bearer. The game is won after one of the teams has caught the flag for a pre-determined number of times. The game objective is not too complex (we know the objective of the other team), making the creation of a motion model a straightforward procedure.

3.2 Tools

The research will make use of the AISandbox, a framework created by the `AIGameDev.com` to challenge people in creating their own AI for a Game of Capture The Flag. The AISandbox offers functionality that takes away the difficulties that are not directly related towards answering my research question, like "are there enemies in my line of sight" or "how do I move through the map", making the problem only of implementing the motion model. The framework also does not give information about the opponent's position, except for the case in which the opponent is directly in the line of sight of one of the teammates, the same assumption as the research question makes.

It would not be efficient to create an AI from scratch for the purpose of this research, so one of the contestants from last competition will be used. This AI, Terminator[1], ended up in third place.

3.3 Motion Modelling

The motion model will be built using Maximum Entropy Inverse Reinforcement Learning. Although implementing it for the purpose of motion modelling has been done before [2] my approach will be different. Their research only used an motion model in the case they had seen the specific opponent played before. This can be improved upon by making multiple prototype models that can be used in case some specific behavior is observed.

The basic idea of meIRL is that the reward function of a Markov Decision Process (which in essence the problem of motion modelling is) has to be learned by observing a set of trajectories belonging to a specific behaviour. By expressing these set of visited positions in features that are map invariant, the reward function can be decided through attaching weights to each of these features and distributing these weights over all positions of the map. Such features are the distance to the bases, the flags and the map center.

3.4 Player classification

To extend the meIRL motion modeller with some prototypical classes instead of using the same model for one player only, a new problem arises. There has to be a way to classify the player's behaviour during the game to know which model to use. It seems most practical to classify not playing-style itself, but typical behaviours that a player can show, making it necessary to switch between different types of motion models during the game. Some behaviours that come to mind for this particular game domain are flag defending, flag attacking and patrolling (where you look for enemies to kill).

The behaviours that could possibly be distinguished need to be modelled by the meIRL-based modeller, but it should also be possible to classify them in real time when playing.

[1]Terminator can be found at `https://github.com/eiisolver?tab=repositories`

To detect these behaviours, a small collection of features are used:

- history of opponent spotted
  - orientation
  - position (expressed in features)
- game state, e.g.
  - Both flags at base
  - Both flags gone
  - Enemy flag intercepted
  - Own flag intercepted

Because the behaviour can change between sightings of the enemy, the difficulty lies in deciding what previous sightings should be considered and which not. I have not yet come up with a solution, except for keeping track of the time of the sightings and ignoring data that is too old.

The best possible classifier still needs to be decided, which probably can be done best when done with collecting the above feature data (so multiple classifiers can be tested at once).

3.5 Reasoning

Having a motion modeller and player classification is not enough, there needs to be a part of the AI that uses the motion prediction to act upon a given situation. Due to time constraints it might be a bit hard to develop the reasoner from scratch, so the Terminator AI will be enhanced with the predictions of the motion model.

Terminator in its current state uses a histogram like approach that maps out all the positions where an enemy has been sighted in the past (of the current game). Using this histogram, safe routes are mapped out for the flag carrier and ambushes are placed. By integrating the meIRL-based motion model with this sighting model, safe paths have a higher chance of being safe (especially when the opponent AI has some adaptive capabilities, as earlier sightings don't give much information about the current game state).

In case there is still time left, it would be a good improvement to enhance the AI with a reasoner that can look further into the future for some anticipating ambushes, but it is highly unlikely there will be time left for this.

4 Evaluation

To make the evaluation answer the research question, different elements of the implementation need to be reviewed. Firstly, it is important that the classifier is working well. We want that given some annotated behaviour, it is possible for the classifier to correctly identify the right kind of behaviour, so that the right prediction model will be applied. To test this, some trajectories (preferably different once than trained with) should be classified.

Because of the way the research question is formulated, some comparison should be made with the works of [2] to conclude if the addition of the general player classification is an improvement. To measure their own success of motion modelling with meIRL, a comparison with a Brownian motion particle filter motion model was made by comparing absolute errors between prediction and real performance. If I repeat the same approach in my own research when measuring absolute error, something meaningful can be said about these results, depending on how much more of an improvement is seen between Brownian motion and meIRL.

A last way with which to evaluate the player classification addition, is by looking at the change in performance of the Terminator AI when the classifier is integrated. By repeating the competition the Terminator took part in and looking at the new win-rates, one can conclude if the meIRL-based classification approach stands a chance in practical situations. Although it might take some time to get the presentation up and running, it is a reasonable possibility as the source code for all competing bots will become available at the end of April.

Evaluating these three parts would give the best idea of the measurable impact of general player classification, but it might be the case that due to time constraints only examining the first two will be achievable.

5 plan

Table 5 shows in short the planning for the upcoming weeks. The planning is based on a weekly schedule, because I do not think planning per day would give a realistic view at all of how my time will be spend (this probably can only be done for one upcoming week at a time, as the planning of the further weeks depends entirely on what you get done in the current week).

At the start of week 18, the main objective will be to create the initial meIRL-based motion model. Two weeks should be enough to gain complete understanding about the mathematics involved, and to implement it. At the end of week 19 I want to make sure the implementation is correct by testing on small situations. At the beginning of week 20 the main focus will lie on creating the classifier. Some of the tasks involved are making sure the necessary features can easily be extracted from a running game, and choosing a method of classification. At the end of the week the decision on a classifier should be definite. With presumably the classifier and motion modeller completed, week 21 is reserved for integrating the classifier into the existing AI code, so that the right predictions are made every time an opponent is observed, and the AI also reasons about predicted data. After week 21 we should have a working AI that given some previous trajectories of opponents can accurately predict the opponents future positions depending on where they were sighted,

Week No.	Research Planning	Report planning
18	Implement meIRL motion model	
19	Implement meIRL motion model	
20	Create classifier	
21	Implement into Terminator AI	
22		Preparation midpresentation and assignment 8
23	Evaluation of classifier, possible adaption	
24	Evaluation of AI performance	Assignment 9
25	Finishing paper	
26	Finishing paper	Preparing final presentation and logbook

and also react on them. The evaluation period of two weeks is first for evaluating the behavior of the classification model. The second week will be used for evaluating the improvement of the performance of the AI with regards to how it functioned before the motion model. The last two weeks are for finishing the report.

References

[1] Stephen Hladky and Vadim Bulitko. An evaluation of models for predicting opponent positions in first-person shooter video games, 2008.

[2] B. Tastan, Yuan Chang, and G. Sukthankar. Learning to intercept opponents in first person shooter games. In *Computational Intelligence and Games (CIG), 2012 IEEE Conference on*, pages 100–107, 2012.

[3] Ben G. Weber, Michael Mateas, and Arnav Jhala. A particle model for state estimation in real-time strategy games. In *Proceedings of AIIDE*, page 103–108, Stanford, Palo Alto, California, 2011. AAAI Press, AAAI Press.