Behavioral Cloning for Simulator Validation

Robert G. Abbott

University of New Mexico, Albuquerque, NM, 87131, USA
rabbott@cs.unm.edu

Abstract. Behavioral cloning is an established technique for creating agent behaviors by replicating patterns of behavior observed in humans or other agents. For pragmatic reasons, behavioral cloning has usually been implemented and tested in simulation environments using a single nonexpert subject. In this paper, we capture behaviors for a team of subject matter experts engaged in real competition (a soccer tournament) rather than participating in a study. From this data set, we create software agents that clone the observed human tactics. We place the agents in a simulation to determine whether increased behavioral realism results in higher performance within the simulation and argue that the transferability of real-world tactics is an important metric for simulator validation. Other applications for validated agents include automated agent behavior, factor analysis for team performance, and evaluation of real team tactics in hypothetical scenarios such as fantasy tournaments.

1 Introduction

Accurate simulation of physical environments and human behavior is important for applications including training [9] and system design concept exploration [3]. However, specific metrics are required for otherwise vague notions of "accuracy" and "realism." For training tactics, a key aspect of accurate simulation is the transferability of tactics between real and simulated environments. If correct tactics are counter-effective in the simulation, students may learn incorrect tactics that work only in the simulator (negative training). We propose a metric for the transferability of tactics: the correlation between agent behavioral realism and agent performance in the simulation.

This approach requires creating software agents with realistic expert behaviors, a task that is often difficult. Domain experts are often unavailable for intensive consulting or are not trained to engineer automated systems. Therefore, domain-specific software is often created by researchers and engineers with only second-hand knowledge of the domain.

Behavioral cloning is one technique to address the challenge of creating agents. Instead of attempting to explain what they know, experts simply perform the task. The performance is recorded, and machine learning algorithms are used to create a model which is used to produce agent behaviors. This paper describes our implementation of behavioral cloning for soccer play and explores the correlation between human model fidelity and performance in the RoboCup

U. Visser et al. (Eds.): RoboCup 2007, LNAI 5001, pp. 329–336, 2008.

soccer simulator. We find a significant correlation, proving that (1) the aspects of behavior captured by the model are significant, (2) the human team employed effective tactics, and (3) real-world tactics are effective within the simulation.

2 Related Work

The goal of software validation is to test completed software for compliance with its specifications. Gledhill and Illgen [6] present a survey of techniques for verification and validation of tactics simulators. For example, *trace validation* is a manual inspection of program state throughout the execution of a test scenario. The techniques are quite general and apply to software for almost any application. Our work focuses on a more specific criterion – tactics validation – that is necessary for tactical trainers.

Behavioral cloning [2] is an established technique for building agent behaviors. A person is observed performing a task and their actions are recorded. A computer model is created to capture patterns of behavior from these observations. The model is then used to control a software agent. Behavioral cloning has been successfully applied in simulations of tasks such as piloting an airplane and operating a crane [10]. Aler, Garcia, and Valls [1] used behavioral cloning in RoboCup to model data collected from a modified version of the simulator that allows a human to play RoboCup as an interactive computer game. However, the captured behavior was a single user manipulating a simulation through computer input devices, rather than a team of soccer players competing in a real match.

For pragmatic reasons, most behavioral cloning research has focused on non-expert subjects (often the researchers themselves) in a computer simulation. Because we are interested in validating simulators against reality, our observations must come from the real world.

3 A Model-Driven Simulated Soccer Team

In selecting an aspect of soccer play for behavior modeling, we considered two criteria: observability (which excludes mental skills such as situational awareness) and transferability from the simulation to reality (which excludes low-level skills such as trapping the ball). We decided to focus on team positioning dynamics.

For low-level skills, our team adopted the UvA Trilearn [7] software library. It is effective, well documented, and open source. We manually implemented a simple kick strategy (including passing, dribbling, and shooting) for our team. Modeling these aspects of play is left for future research.

3.1 A Data Set of Human Soccer Play

Our research needed a data set of skilled human soccer play. Creating the data set required both source material (recorded soccer play) and a tool to extract

data from the source material. Televised soccer footage fails to capture team positioning dynamics because it focuses on the ball, so we recorded games at the University of New Mexico soccer pitch. A single video camera was unable to record the entire field with adequate resolution, so we used an array of four cameras along the top row of bleachers overlooking the field 1.

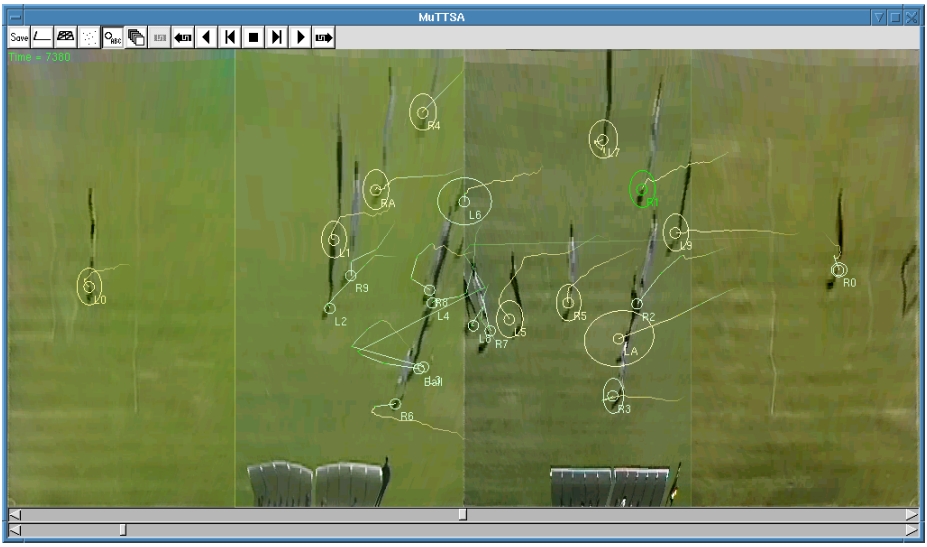


Fig. 1. A screenshot from the motion capture application. The application shows an overhead view synthesized from four cameras. Rings and trails indicate the players' current and recent positions.

Our data extraction tool is a multiple target tracking application that fuses the four viewpoints, detects players using background subtraction [5], and tracks players using Joint Probability Data Association [8] with Kalman Filtering.

The resulting dataset consists of a sequence of observations of the state of the game. Each observation consists of the positions of all 22 players (11 per team) and the ball at a sample rate of $10Hz$. Other information (e.g., velocities and distances) is derived from the sequence of positions.

3.2 Cloning Team Positioning Dynamics

We modeled team positioning dynamics using a function approximator to predict the position of each human player given the current state of the environment (e.g., the positions of players and the ball). We used nearest-neighbor matching, which is a type of instance-based learning (IBL). In the soccer domain, each instance is an observation of the state of the soccer game at a moment in time. In our implementation, an observation is an associative array of named features.

Each feature may be multidimensional. Many of the features are two dimensional because we used the 2d version of the RoboCup simulator.

3.3 Clustering and Model Context Set Size

The basic nearest-neighbor algorithm is highly susceptible to over-fitting. The model will perform poorly in the regions of nonrepresentative instances. We used clustering to prune the instance set by discarding redundant and non-representative instances; only the cluster centers are retained. We refer to retained instances as contexts.

The use of clustering adds a parameter (the number of contexts) to the model. The impact of this parameter on predictive accuracy is measured in Section 4. The number of contexts also determines the run-time memory and computation requirements of the model, which is especially important for team-oriented tasks.

3.4 Distance Metric and Feature Selection

In a complex task such as soccer, a team will never encounter precisely the same situation twice, so it is crucial to draw correct analogies between the current situation and relevant model contexts. Relevance is defined by a distance function. Our implementation calculates the weighted Euclidean distance between observation vectors. The weight vector is a parameter to the system and must be chosen to emphasize the features that most influence expert tactics. A weight of 0 causes the corresponding feature to be disregarded entirely. Feature selection algorithms [4] could be used automate the selection of weights.

Feature selection is critical for IBL, which is intolerant of irrelevant attributes. We obtained the best results for soccer field positioning with a small set of carefully chosen features as described in Section 4.

3.5 Number of Training Observations

The success of behavioral cloning for tactics modeling is indicated by the sensitivity of the model to varying amounts of training data. If varying the amount of training data has no effect, then there is no transfer of human expertise to the system, and the human model is worthless. If performance increases very slowly with increasing amounts of training data, then gathering enough data to yield a substantial performance improvement may be prohibitively labor intensive.

The feature set, number of model contexts, and amount of training data are all interdependent parameters. For example, a larger amount of training data contains more varied situations and behavior and may require more features and contexts to model effectively.

4 Experiments in Behavioral Cloning for RoboCup

This section describes two experiments. The first measures the predictive accuracy of our player-positioning model on the human data set. The second measures

the outcome of RoboCup matches when the model is used to control RoboCup agents. These provide the information required to calculate the correlation between behavior modeling accuracy and agent performance in the simulation.

4.1 Predictive Accuracy of Human Player Model

The first experiment determined how prediction accuracy varies with increasing amounts of training data. We produced a data set of human soccer play by capturing the first 20 minutes of a soccer match between the University of California Irvine Anteaters and the Western Illinois University Leathernecks at the University of New Mexico Soccer Complex on September 21, 2003. This data set is available at `http://www.cs.unm.edu/~rabbott/SoccerData`

Results are reported for three feature sets: "Ball Position X" uses only the component of ball position extending from one goal to the other; "Ball Position" contains the 2-dimensional ball position; "Ball Position and Velocity" includes both the ball's position and estimated velocity. We experimented with larger feature sets and more complex features (such as the density of opponents between a player and the ball) but without significant improvement in the results.

We used 10-fold cross-validation to measure predictive accuracy. For each fold, we measured the mean prediction error (the squared distance between predicted and observed player positions in holdout data) for every combination of observation set size, context set size, and selected features.

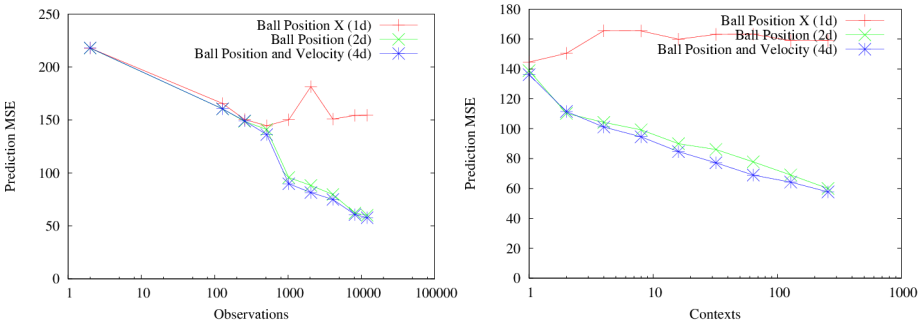
4.2 Predictive Accuracy of Human Player Model – Results

Figure 2 shows how feature selection, observation set size, and context set size influence prediction accuracy for the human soccer data set. These results show that the x component of ball position is insufficient to accurately predict player position. In contrast, the Ball Position and Ball Position and Velocity feature sets achieve lower prediction error, with steadily improving results up to all available observations (Figure 2(a)) and the maximum tested number of contexts (Figure 2(b)). The inclusion of ball velocity yields an improvement that is very small, but consistent across varying observation and context set sizes.

4.3 RoboCup Performance

All simulations have limited fidelity. Even a perfect model of ideal human behavior on a task might not perform well in a simulation of the same task because of differences between the model and the real world. We evaluated the performance of our model-driven RoboCup team using the same parameterizations as in the previous section. The opponent team is a behavior clone of the UvA Trilearn RoboCup team [7]. Training data for this clone was generated by running several simulation matches between two instances of the UvA Trilearn.

The performance metric is the goal difference $PenaltyScore = Goals_{opponent} - Goals_{self}$ during a five minute match. As with the error metric used in the previous section, a lower penalty score indicates better performance. (The penalty



(a) Prediction error for each observation set size.

(b) Prediction error for each context set size.

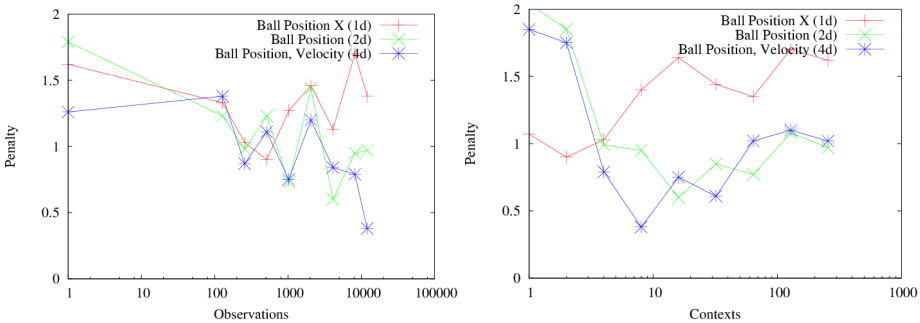
Fig. 2. Summary of results from the prediction error parameter sweep

score reflects only the outcome of a match and is not related to penalties assessed by a referee for violating the rules of soccer).

As before, we tested all combinations of three feature sets and nine values each for the observation set and context set sizes. For each condition, we computed the mean of 100 trials for a total of 24,300 RoboCup matches.

4.4 RoboCup Performance – Results

The RoboCup performance for all conditions is summarized in Figure 3. The performance of Ball Position X is maximized with only 512 observations (about 50*s* of play) and decreases with additional observations. With this simple feature set, the model cannot adequately distinguish between different situations



(a) Penalty for each observation set size.

(b) Penalty for each context set size.

Fig. 3. Summary of results from the RoboCup performance parameter sweep

in soccer. For the other two feature sets, performance generally improves with additional observations, and the maximum performance is achieved using both ball position and velocity with all available observations.

The result of varying the context set size is quite different; for all three feature sets, performance declines when more than 16 contexts are created (Figure 3(b)), even as the human prediction accuracy continues to increase (Figure 2(b)). This seems to indicate that clustering is an effective technique to prevent over-fitting when using IBL.

Figure 4 displays the correlation between model prediction error on the human soccer dataset and the penalty score for a RoboCup team controlled by the same model. This calculation is important because it tests the hypothesis that real soccer strategy is effective within the RoboCup simulator. No significant correlation is found for Ball Pos X ($r = 0.07$), perhaps because of the relatively small range of prediction accuracy observed for this feature. However, a significant correlation exists for Ball Position ($r = 0.41$) and Ball Position and Velocity ($r = 0.50$). When all conditions are taken together, the overall correlation is 0.43.

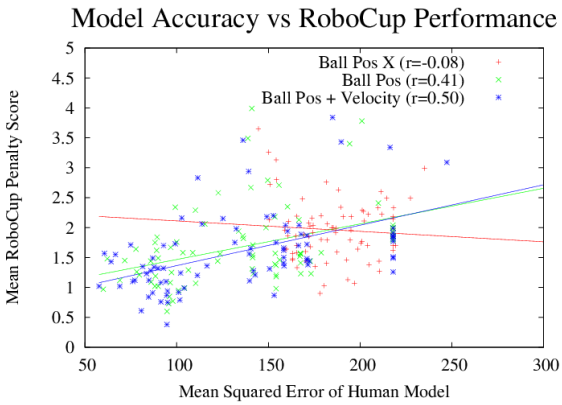


Fig. 4. Correlation between model predictive accuracy on the human soccer dataset and performance of a model-driven RoboCup team. Each data point represents a model with a unique set of parameter values. There are three point clouds, each representing a different feature set.

5 Discussion and Conclusion

The correlation between human model fidelity and performance in the simulation ($r = 0.43$) is significant, confirming the hypothesis that human soccer strategies are effective within RoboCup. What factors account for this correlation and at the same time prevent a stronger correlation? We propose three factors.

First, the correlation between predictive accuracy and performance is limited by the significance of the behaviors captured by the model. Our model captures team field positioning strategy, but does not capture other important factors such as pass selection and individual ball handling skill. In pedagogy, there is a risk of focusing on unimportant knowledge and skills simply because, for example, they are easy to explain or were important historically. The relative importance of various skills could be studied by modeling each and then measuring the impact of degrading one or more of the models.

Second, all humans (and human teams) are imperfect to varying degrees. Emulating a team of novices *should* result in worse performance than emulating a World Cup match. Thus, it may be possible to predict the outcome of a match through a contest of behavior clones.

Third, all simulations fall short of perfect realism. Ideal RoboCup tactics are distinct from ideal soccer tactics. This is a problem for training simulators because students will be discouraged from practicing proven tactics if they are ineffective in the simulator. Simulation developers should strive for a high correlation between desirable behavior and positive scenario outcomes.

Creating tests to isolate these factors is an important topic for future research. Such tests would allow instructors to re-target training, assess student performance, or focus on increasing simulator fidelity. However, any significant correlation between expert cloning accuracy and agent performance demonstrates that the modeled behavior is significant, that the example behavior is skilled, and that real-world tactics are transferable to the simulation.

References

1. Aler, R., Garcia, O., Valls, J.M.: Correcting and improving imitation models of humans for robosoccer agents. In: The 2005 IEEE Congress on Evolutionary Computation, vol. 3, pp. 2402–2409 (2005)
2. Bratko, I., Urbancic, T., Sammut, C.: Machine Learning and Data Mining: Methods and Applications. In: Behavioural Cloning of Control Skill. John Wiley & Sons Ltd. (1997)
3. Cares, J.R.: Agent modeling: the use of agent-based models in military concept development. In: WSC 2002, pp. 935–939 (2002)
4. Cost, S., Salzberg, S.: A weighted nearest neighbor algorithm for learning with symbolic features. Machine Learning 10, 57–78 (1993)
5. Fuentes, L.M., Velastin, S.A.: People tracking in surveillance applications. In: PETS 2001, Hawaii (2001)
6. Gledhill, D.W., Illgen, J.D.: 21st century verification and validation techniques for synthetic training models and simulations. In: Proc. I/ITSEC (1999)
7. Kok, J.R., Boer, R.: The incremental development of a synthetic multi-agent system: the UvA Trilearn 2001 robotic soccer simulation team. Master's thesis. University of Amsterdam, The Netherlands (2002)
8. Oh, S., Sastry, S.: A polynomial-time approximation algorithm for joint probabilistic data association. In: Proc. Am. Control Conf. (2005)
9. Stone, B.: Serious gaming. Defence Management Journal (2005)
10. Suc, D., Bratko, I.: Symbolic and qualitative reconstruction of control skill. Electronic transactions on artificial intelligence, Section B 3, 1–22 (1999)