Opponent Modeling and Spatial Similarity to Retrieve and Reuse Superior Plays

Kennard Laviers
School of EECS
U. of Central Florida
Orlando, FL
klaviers@eecs.ucf.edu

Gita Sukthankar
School of EECS
U. of Central Florida
Orlando, FL
gitars@eecs.ucf.edu

Matthew Klenk
NCARAI
Naval Research Lab
Washington, DC
matthew.klenk.ctr@nrl.navy.mil

David W. Aha
NCARAI
Naval Research Lab
Washington, DC
david.aha@nrl.navy.mil

Matthew Molineaux
Knexus Research
Springfield, VA
matthew.molineaux@knexusresearch.com

Abstract

By analyzing play history, it is possible to gain critical insights about future plays. Plays are sequences of actions to be undertaken by a collection of agents, or teammates. The success of a play depends on a number of factors including, perhaps most importantly, the opponent's play. In this paper, we present an approach for online opponent modeling and illustrate how it can be used to improve offensive performance in the Rush 2008 football simulator. In football, team behaviors have an observable spatio-temporal structure, defined by the relative physical positions of team members over time. We demonstrate that this structure can be exploited to recognize football plays at a very early stage of the play using a supervised learning method. Using the recognized defensive play, knowledge about expected outcomes, and spatial similarity between offensive plays, we retrieve an offensive play from case base. This play is then reused to improve an in progress offensive play. We call this process a *play switch*. Empirical results indicate that spatial similarity is central to play retrieval, and that, modifying only a subset of the current play with the retrieved play yields greater improvement.

Introduction

To succeed at American Football, a team must be able to successfully execute closely-coordinated physical behavior. Teams rely upon a pre-existing sets of offensive and defensive plays, or *playbooks*, to achieve this coordinated behavior. By analyzing play history, it is possible to glean critical insights about future plays. In American Football, quarterbacks frequently call *audibles*, changes of play based on an assessment of the opponent's play. This task involves identifying the opponent's play and then selecting a new play for the offensive team.

In physical domains (military or athletic), team behaviors often have an observable spatio-temporal structure, defined by the relative physical positions of team members. This structure can be exploited to perform behavior recognition on traces of agent activity over time. This paper describes a method for recognizing defensive plays from spatio-temporal traces of player movement in the Rush 2008 Football Simulator (Figure 1). Rush 2008 simulates a modified version of American Football and was developed from the open source Rush 2005 game (Rush 2005), which is similar in spirit to Tecmo Bowl and NFL Blitz.

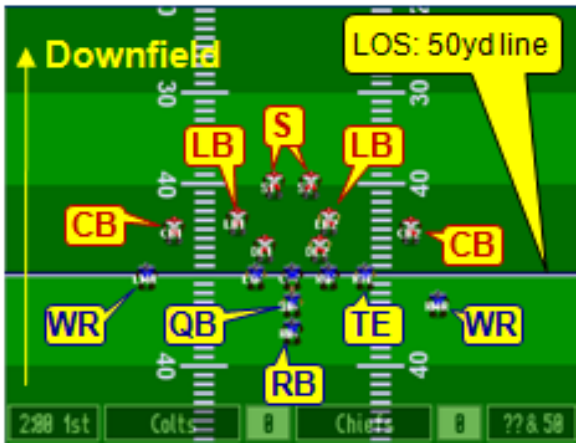

Figure 1: Screenshot of the Rush 2008 football simulator.

Using knowledge of play histories, we present a method for executing a *play switch* based on the potential of other plays to improve the yardage gained and their similarity to the current play. From a case based reasoning perspective (Aamodt and Plaza 1994), this involves retrieving a superior play and adapting it to the current situation. In retrieving a superior play, we show that calculating the relative similarity of the current play compared with the candidate play improves performance. By limiting the play switch to a subgroup of players, we can improve on the total team switch.

We begin by describing the Rush Football simulator. Next we describe our play switching approach with a detailed discussion of opposing play recognition, play similarity, and play adaptation. We outline the system which implements these ideas and present an empirical evaluation. We close with related and future work.

Rush Football

Football is a contest of two teams played on a rectangular field that is bordered on lengthwise sides by an end zone. Unlike American Football, Rush teams only have 8 players on the field at a time out of a roster of 18 players. The field is 100 yards by 63 yards. The game's objective is to outscore the opponent, where the offense (i.e., the team with possession of the ball), attempts to advance the ball from the

line of scrimmage into their opponent's end zone. Therefore, an offensive play's success can be measured by the amount of yardage gain.

The offensive lineup contains the following positions:

Quarterback (QB): given the ball at the start of each play, and will initiate either a run or a pass.

Running back (RB): begins in the backfield, behind the line of scrimmage where the ball is placed, with the quarterback and fullback. The running back is eligible to receive a handoff, pitch or pass from the quarterback.

Full back (FB): serves largely the same function as the running back.

Wide receiver (WR): executes passing routes and is the primary receiver for pass plays. The wide receiver initially starts near the line of scrimmage but on the far right or far left of the field.

Tight end (TE): begins on the line of scrimmage immediately to the outside of the offensive lineman and can receive passes.

Offensive lineman (OL): starts on the line of scrimmage and is primarily responsible for preventing the defense from reaching the ball carrier.

A Rush play is composed of (1) a starting formation and (2) instructions for each player in that formation. A formation is a set of (x,y) offsets from the center of the line of scrimmage. By default, instructions for each player consist of (a) an offset/destination point on the field to run to, and (b) a behavior to execute when they get there. Play instructions are similar to a conditional plan and include choice points where the players can make individual decisions as well as pre-defined behaviors that the player executes to the best of their physical capability. Rush includes three offensive formations (power, pro, and split) and four defensive ones (23, 31, 2222, 2231). Each formation has eight different plays (numbered 1-8) that can be executed from that formation. Offensive plays typically include a handoff to the running back/fullback or a pass executed by the quarterback to one of the receivers, along with instructions for a running pattern to be followed by all the receivers. An example play is given below:

- the quarterback will pass to an open receiver;
- the running back and fullback will run hook routes;
- the left wide receiver will run a corner right route;
- the right wide receiver will run a hook route;
- the other players will block for the ball holder.

Offensive Play Switches

In American Football, the quarterback changes the play based on the defensive formation and their reactions to offensive actions before the beginning of the play. Although Rush does not allow for actions before the play, the Rush simulator allows us to alter the play shortly after it has begun, providing an analogous task.

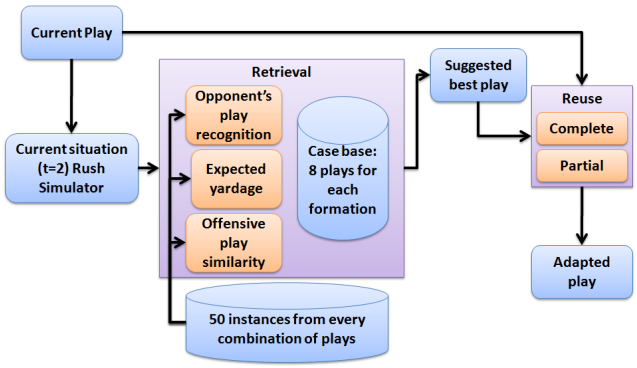

Figure 2: Play-switching approach.

Our approach focuses on two aspects of case-based reasoning, retrieval and reuse (Aamodt and Plaza 1994). At this early stage, we are not concerned with the revision or retention of play-switching episodes for future use. Our play switch approach is summarized in Figure 2. Play retrieval requires quickly recognizing the opponent's play, predicting the results of different offensive plays against it, and comparing them to the current situation. This retrieved play is reused by giving new actions to players in the current situation. Our case base consists of the 8 plays for each of the three offensive formations.

The system's background knowledge includes 50 instances of every offensive and defensive play combination. These instances are used to train the recognition system, generate an expected yardage table for every combination of plays, and compute similarity between the offensive plays. The next sections describe the play recognition and similarity metric used in retrieval, followed by a discussion of how the retrieved play is adapted for the current situation.

Play Recognition using SVMs

Given a series of observations, our goal is to recognize the defensive play as quickly as possible in order to maximize our team's ability to intelligently respond with the best offense. Thus, the observation sequence grows with time unlike in standard offline activity recognition where the entire set of observations is available. We approach the problem by training a series of multi-class discriminative classifiers, each of which is designed to handle observation sequences of a particular length. In general, we expect that the early classifiers will be less accurate since they are operating with a shorter observation vector and because the positions of the players have deviated little from the initial formation.

We perform this classification using support vector machines (Vapnik 1998). Support vector machines (SVM) are a supervised algorithm that can be used to learn a binary classifier; they have been demonstrated to perform well on a variety of pattern classification tasks, particularly when the dimensionality of the data is high (as in our case). Intuitively an SVM projects data points into a higher dimensional space, specified by a kernel function, and computes a maximum-margin hyperplane decision surface that sepa-

rates the two classes. Support vectors are those data points that lie closest to this decision surface; if these data points were removed from the training data, the decision surface would change. More formally, given a labeled training set $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_l, y_l)\}$, where $\mathbf{x}_i \in \Re^N$ is a feature vector and $y_i \in \{-1, +1\}$ is its binary class label, an SVM requires solving the following optimization problem:

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i$$

constrained by:

$$\begin{aligned} y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + \mathbf{b}) &\geq 1 - \xi_i, \\ \xi_i &\geq 0. \end{aligned}$$

The function $\phi(.)$ that maps data points into the higher dimensional space is not explicitly represented; rather, a *kernel* function, $K(\mathbf{x}_i, \mathbf{x}_j) \equiv \phi(\mathbf{x}_i)\phi(\mathbf{x}_j)$, is used to implicitly specify this mapping. In our application, we use the popular radial basis function (RBF) kernel:

$$K(x_i, x_j) = \exp(-\gamma||x_i - x_j||^2), \gamma > 0.$$

Several extensions have been proposed to enable SVMs to operate on multi-class problems (with k rather than 2 classes), such as one-vs-all, one-vs-one, and error-correcting output codes. We employ a standard one-vs-one voting scheme where all pairwise binary classifiers, $k(k-1)/2 = 28$ for every multi-class problem in our case, are trained and the most popular class is selected. Many efficient implementations of SVMs are publicly available; we use LIBSVM (Chang and Lin 2001).

We train our classifiers using a collection of simulated games in Rush collected under controlled conditions: 40 instances of every possible combination of offense (8) and defense plays (8), from each of the 12 starting formation configurations. Since the starting configuration is known, each series of SVMs is only trained with data that could be observed starting from its given configuration. For each configuration, we create a series of training sequences that accumulates spatio-temporal traces from $t = 0$ up to $t \in \{2, \ldots, 10\}$ time steps. A multiclass SVM (i.e., a collection of 28 binary SVMs) is trained for each of these training sequence lengths. Although the aggregate number of binary classifiers is large, each classifier only employs a small fraction of the dataset and is therefore efficient (and highly parallellizable). Cross-validation on a training set was used to tune the SVM parameters (C and σ) for all of the SVMs.

Classification at testing time is very fast and proceeds as follows. We select the multiclass SVM that is relevant to the current starting configuration and time step. An observation vector of the correct length is generated (this can be done incrementally during game play) and fed to the multi-class SVM. The output of the intent recognizer is the system's best guess (at the current time step) about the opponent's choice of defensive play and can help us to select the most appropriate offense, as discussed below.

Table 1 summarizes the experimental results for different lengths of the observation vector (time from start of play),

Table 1: Play recognition results (all play combinations)

$t = 2$	3	4	6	8	10
12.50	96.88	96.87	96.84	96.89	96.81

averaging classification accuracy across all starting formation choices and defense choices. We see that at the earliest timestep, our classification accuracy is at the baseline but jumps sharply near perfect levels at $t = 3$. This strongly confirms the feasibility of accurate intent recognition in Rush, even during very early stages of a play. At $t = 2$, there is insufficient information to discriminate between defensive plays (perceptual aliasing), however by $t = 3$, the positions of the defensive team are distinctive enough to be reliably recognized. Thus, for our agent, we use $t = 3$ to classify the opposing play.

Play Similarity Metric

While knowledge about the opposing play is central to retrieving an effective offensive play, the similarity of the candidate plays to the current play estimates the feasibility of the play switch.

To calculate play similarities, we create a feature matrix for every formation/play combinations based on background knowledge. The 13 features for each athlete A include max, min, mean, and median over x and y in addition to the following special features:

FirstToLastAngle: Angle from starting point (x_0, y_0), to ending point (x_n, y_n), defined as $atan\left(\frac{\triangle y}{\triangle x}\right)$

StartAngle: Angle from the starting point (x_0, y_0) to (x_1, y_1), defined as $atan\left(\frac{y_1 - y_0}{x_1 - x_0}\right)$

EndAngle: Angle from (x_{n-1}, y_{n-1}) to the ending point (x_n, y_n), defined as $atan\left(\frac{\triangle y}{\triangle x}\right)$

TotalAngle: $\sum_{i=0}^{N-1} atan\left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i}\right)$

TotalPathDist: $\sum_{i=1}^{N} \sqrt[2]{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}$

These features are similar to the ones used in (Rubine 1991) and more recently by (Wobbrock *et al.* 2007) to match pen trajectories in sketch-based recognition tasks, another spatio-temporal task. Here, they are generalized for use with multi-player trajectories. Feature set F for a given play c ($c = 1...8$, represents possible play matches per formation) contains all features for each offensive player in the play and is described as:

$$\overrightarrow{F_c} = \{A_{c1} \cup A_{c2} \cup \cdots \cup A_{c8}\}$$

Using the 50 play instances from background knowledge, we compute a similarity vector V for every combination of offensive formation, offensive play, defensive formation, and defensive play combination. This vector includes 8 entries (the computed similarities between the offensive play and the other plays from that formation). We define the similarity between plays as the sum of the absolute value

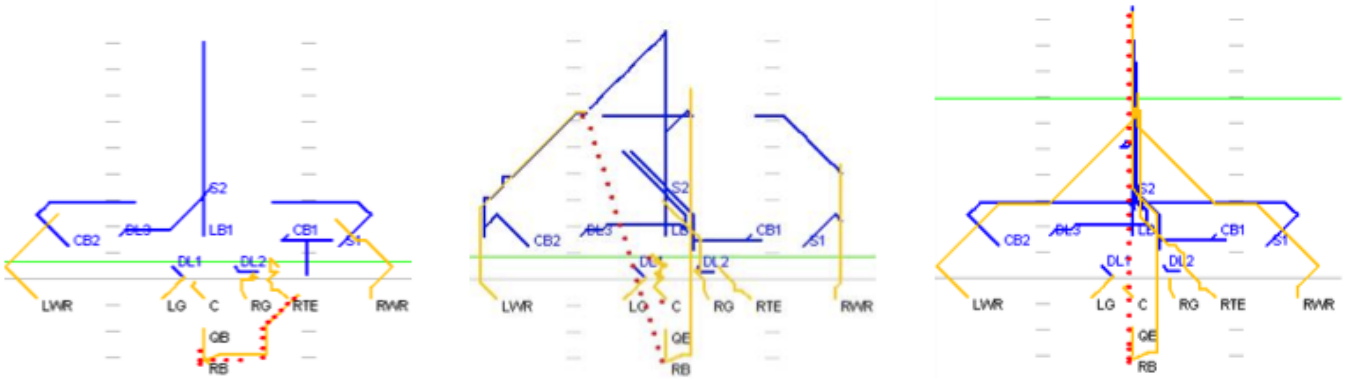

Figure 3: The starting play (left) is played out, first by changing all players to the optimal play (center) and then by executing change commands only for a subgroup of players (right). Effectively, the subgroup switch play remains close the original play by player movements while becoming significantly more effective as indicated by the green line which represents average yardage gained in this play.

of the differences (L_1 norm) between features F_{c_i} and F_{c_j}. In the evaluation section, we compare the performance of a similarity-based play switch mechanism vs. a play switching algorithm that focuses solely on the potential for yardage gained.

Play Reuse

To reuse the new play in the current situation, we must adapt the current play. The most straightforward approach involves changing the entire play (i.e., each offensive player follows the new play from this time forward). An alternative strategy involves modifying the actions of only a small group of key players while leaving others alone. By segmenting the team in this fashion, we are able to combine two plays that had previously been identified as alike with regard to spatio-temporal data, but different in regards to yards gained. Based on our domain knowledge of football, we selected three subgroups as candidates to switch: {QB, RB, FB}, {LG, C, RG}, and {LWR, RWR, RTE, LTE}.

Figure 3 displays a good example of a successful merging of two plays to produce a superior play with subgroup switching. The green line represents the average yardage gained. The left image is the most likely path of the baseline case (a running play which yields little yardage on average). The middle image is the most likely execution trace produced by the total play switch method. The play produced by the total play switch was not much more successful than the baseline case. However, when Group 1 {QB, RB, FB} is modified, the yardage gained increased greatly and the new play is shown to be very coordinated and effective.

Improving the Offense with Play Switches

To improve offensive performance, our agent evaluates the competitive advantage of executing a play switch based on 1) the potential of other plays to improve the yardage gained and 2) the similarity of the candidate plays to the current play. Our algorithm for improving Rush offensive play has two main phases: a preprocess stage, which yields a play switch lookup table, and an execution stage, where the defensive play is recognized and the offense responds with an appropriate play switch for that defensive play. We train a set of SVM classifiers using 40 instances of every possible combination of offense (8) and defense plays (8), from each of the 12 starting formation configurations. This stage yields a set of models used for play recognition during the game. Next, we calculate and cache play switches using the following procedure:

1. Collect data by running the RUSH 2008 football simulator 50 times for every play combination.
2. Create yardage lookup tables for each play combination. This information alone is insufficient to determine how good a potential play is to perform the play switch action on. The transition play must resemble our current offensive play or the offensive team will spend too much time retracing steps and perform very poorly.
3. Compute similarity matrix between offensive plays for all formation/play combinations.
4. Create the final play switch lookup table based on both the yardage information and the play similarity.

To create the play switch lookup table, the agent first extracts a list of offensive plays L given the requirement $yards\,(L_i) > \epsilon$ where ϵ is the least amount of yardage gained before the agent changes the current offensive play to another. We used $\epsilon = 1.95$ based on a quadratic polynomial fit of total yardage gained in 6 tests with $\epsilon = \{MIN, 1.1, 1.6, 2.1, 2.6, MAX\}$ where MIN is small enough no plays are selected to change and MAX where all plays are selected for change to the highest yardage play with no similarity comparison. Second, from the list L find the play most similar to our current play, and add it to the lookup table.

During execution, the offense uses the following procedure:

1. At each observation less than 4, collect movement traces for each play.

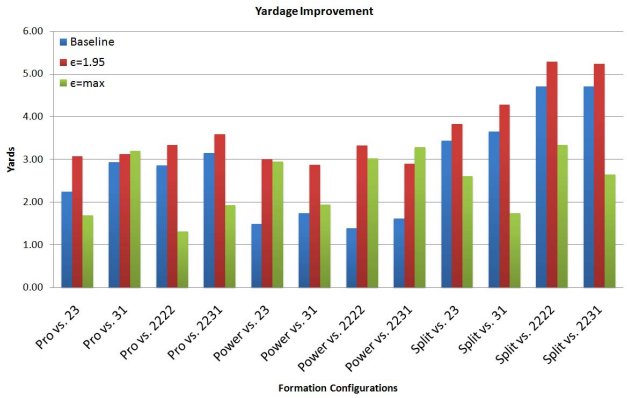

Figure 4: Comparison of greedy play switch and similarity-based switching. Our similarity-based play switch method (shown in red) outperforms both baseline Rush offense (blue) and a greedy play switch metric (green).

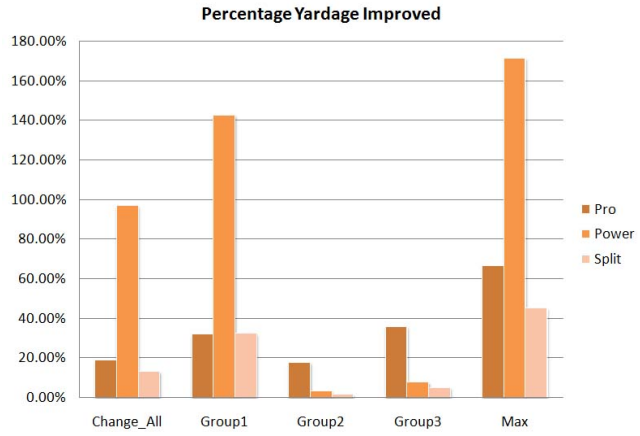

Figure 5: The play-yardage gain over baseline Rush offense yielded by various play switch strategies.

2. At observation 3, use LIBSVM with the collected movement traces and previously trained SVM models to identify the defensive player.
3. Access the lookup table to find $best(i)$ for our current play i.
4. If $best(i) \neq i$, Send a change order command to the offensive team to change to play $best(i)$.

As described in the Adapting the Current Play section, our system allows for different methods of using the retrieved play. The agent can switch the play for either every offensive player or a subset.

Empirical Evaluation

Our goal is the answer the following questions:

1. Does our play switching algorithm improve yardage gained?
2. Does retrieval incorporating similarity with the current play outperform a greedy strategy which selects solely based upon expected yardage gained?
3. What are the effects of subgroup switching on play performance?

To answer the first two questions, we ran the RUSH 2008 simulator for ten plays on each possible play configuration under three conditions: a baseline without any play switching, our play switch model (using the yardage threshold $\epsilon = 1.95$ as determined by the quadratic fit), and a greedy play switch strategy based solely on the yardage table ($\epsilon = MAX$). The results are shown in Figure 4.

Overall, the average performance of the offense went from 2.82 yards per play to 3.65 yards per play ($\epsilon = 1.95$) with an overall increase of 29%, $\pm 1.5\%$ based on sampling of three sets of ten trials. An analysis of each of the formation combinations (Figure 4) shows the yardage gain varies from as much as 100% to as little as 0.1%. Power vs. 23 is dramatically boosted from about 1.5 yards to about 3 yards per play, doubling yards gained. Other combinations, such as Split vs. 23 and Pro vs. 32 already gained high yardage and improved less dramatically at about .2 to .4 yards more than the gains in the baseline sample. Overall, our model's performance is consistently better for every configuration tested.

Results with $\epsilon = MAX$ clearly shows simply changing to the greatest yardage generally results in poor performance from the offense. When the similarity metric is not used, the results are drastically reduced. The reason appears to be mis-coordinations between teammates accidentally induced by the play switch; by maximizing the play similarity simultaneously, the possibility of mis-coordinations is reduced.

To evaluate the subgroup switching, we ran the simulation three additional trails. In each trial, our play switching method was allowed to switch only one of the offensive player subgroups. Using the improvement in yardage, we compared these trials to the full offense switch and the best offensive play against the defense. The results (shown in Figure 5) clearly indicated the best subgroup switch (consistently Group 1) produced greater gains than the total team switch, which still performed better than the baseline. Early play recognition combined with subgroup switching yields the best results, assuming no oracular knowledge of the other team's intentions prior to run-time.

Related Work

Previous work on team behavior recognition has been primarily evaluated within athletic domains, including American Football (Intille and Bobick 1999), basketball (Bhandari *et al.* 1997; Jug *et al.* 2003), and Robocup soccer simulations (Riley and Veloso 2000; 2002; Kuhlmann *et al.* 2006). In Robocup, most of the research on team intent recognition focused on coaching. Techniques have been developed to extract specific information, such as home areas (Riley *et al.* 2002), opponent positions during set-plays (Riley and Veloso 2002), and adversarial models (Riley and Veloso 2000), from logs of Robocup simulation league games. However, the coaching agents use offline

processing to improve their team's performance in future games. In contrast, our agent immediately takes action on the recognized play to evaluate possible play switches.

Comparatively few case-based reasoning researchers have investigated spatial reasoning. Most focus on retrieving precedents based on quantitative and qualitative features (Holt and Benwell 1995) without any adaption. Using insights from research on pen stroke recognition (Wobbrock *et al.* 2007), our spatial similarity metric incorporates spatio-temporal knowledge into retrieval, which is then used to adapt the current situation. Galatea (Davies *et al.* 2005) uses stored visual problem-solving episodes consisting of visual transformations, which are employed analogically to arrive at a solution for new problems. While transfer in Galatea is iterative, our play switch is a one-shot process. Furthermore, Galatea places little emphasis on retrieval. Our model uses spatial knowledge throughout retrieval, first in categorizing the opposing teams play, then in determining the most similar play from the case base.

Rush 2008 was developed as a platform for evaluating game-playing agents and has been used to study the problem of learning strategies by observation (Li *et al.* 2009). Intention recognition has been used within Rush 2008 as part of a reinforcement learning method for controlling a single quarterback agent (Molineaux *et al.* 2009). In this paper, our approach addresses policies across *multiple* agents.

Conclusion

Accurate opponent modeling is an important stepping-stone toward the creation of interesting autonomous adversaries. In this paper, we present an approach for online strategy recognition in the Rush 2008 football simulator. After identifying the defense's play, our agent evaluates the advantage of executing a play switch based on the potential of other plays to improve the yardage gained and their similarity to the current play.

We have shown that spatio-temporal features enable online strategy recognition in the early stages of a play. Furthermore, by incorporating spatial similarity into the selection of the appropriate play switch, our method avoids mis-coordinations between offensive players, increasing the yardage gained. Additionally, we demonstrate that limiting the play switch to a subgroup of key players further improves performance.

In future work, we plan on extending our game playing agent to play the entire game. While our focus on gaining more yards is central to successful offense, in the complete game, offensive strategy becomes more complex, including scoring and clock management. As discussed previously, we plan to explore methods for automatically identifying key player subgroups for adapting the play by examining motion correlations between players. Finally, we plan to explore these ideas of online strategy recognition in other domains.

References

A. Aamodt, E. Plaza. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI Communications*. IOS Press, Vol. 7: 1, pp. 39-59. 1994.

I. Bhandari, E. Colet, J. Parker, Z. Pines, R. Pratap, and K. Ramanujam. Advanced Scout: Data mining and knowledge discovery in NBA data. *Data Mining and Knowledge Discovery*, 1(1):121–125, 1997.

C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

J. Davies, A. Goel and N. Nersessian. Transfer in visual case-based problem-solving. In H. Munoz-Avila & F. Ricci (Eds.) *Proceedings of the 6th International Conference on Case-Based Reasoning*. Springer-Verlag. Berlin Heidelberg. 2005.

A. Holt and G. Benwell. Case-based reasoning with spatial data. In *2nd New Zealand Two-Stream International Conference on Artificial Neural Networks and Expert Systems (ANNES '95)*, 1995.

S. Intille and A. Bobick. A framework for recognizing multi-agent action from visual evidence. In *Proceedings of National Conference on Artificial Intelligence*, 1999.

M. Jug, J. Pers, B. Dezman, and S. Kovacic. Trajectory based assessment of coordinated human activity. In *Proceedings of the International Conference on Computer Vision Systems (ICVS)*, 2003.

G. Kuhlmann, W. Knox, and P. Stone. Know thine enemy: A champion RoboCup coach agent. In *Proceedings of National Conference on Artificial Intelligence*, 2006.

N. Li, D. Stracuzzi, G. Cleveland, P. Langley, T. Konik, D. Shapiro, K. Ali, M. Molineaux, and D. Aha. Constructing game agents from video of human behavior. In *Proceedings of IJCAI Workshop on Learning Structural Knowledge from Observations*, 2009.

M. Molineaux, D. Aha, and G. Sukthankar. Beating the defense: Using plan recognition to inform learning agents. In *Proceedings of Florida Artifical Intelligence Research Society*, 2009.

P. Riley and M. Veloso. On behavior classification in adversarial environments. In L. Parker, G. Bekey, and J. Barhen, editors, *Distributed Autonomous Robotic Systems 4*. Springer-Verlag, 2000.

P. Riley and M. Veloso. Recognizing probabilistic opponent movement models. In A. Birk, S. Coradeschi, and S. Tadorokoro, editors, *RoboCup-2001: Robot Soccer World Cup V*. Springer Verlag, 2002.

P. Riley, M. Veloso, and G. Kaminka. An empirical study of coaching. In H. Asama, T. Arai, T. Fukuda, and T. Hasegawa, editors, *Distributed Autonomous Robotic Systems 5*. Springer-Verlag, 2002.

D. Rubine. Specifying gestures by example. *Computer Graphics, Volume 25, Number 4*, pages 329–337, 1991.

Rush, 2005. `http://sourceforge.net/projects/rush2005/`.

V. Vapnik. *Statistical Learning Theory*. Wiley & Sons, Inc, 1998.

J. Wobbrock, D. Wilson, and L. Yang. Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. In *Symposium on User Interface Software and , Proceedings of the 20th annual ACM symposium on User interface software and technology*, 2007.