Exploiting Early Intent Recognition for Competitive Advantage

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

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

Abstract

In physical domains (military or athletic), team behaviors often have an observable spatio-temporal structure, defined by the relative physical positions of team members over time. In this paper, we demonstrate that this structure can be exploited to recognize football plays in the Rush 2008 football simulator. Although events in the simulator are stochastically generated, we present a method for reliably recognizing football plays at a very early stage using multiple support vector machines; moreover, we demonstrate that having this early information about the defense's intent can be utilized to improve offensive team play. Our system evaluates the competitive advantage of executing a *play switch* based on the potential of other plays to improve the yardage scored and the similarity of the candidate plays to the current play. Our play switch selection mechanism outperforms both the basic offense and a greedy yardage-based switching strategy.

1 Introduction

This paper addresses the problem of early recognition of opponent intent in adversarial team games. 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. There are three general types of cues that can be used to perform recognition:

- **spatial** relationships between team members and/or physical landmarks that remain fixed over a period of time;
- **temporal** dependencies between behaviors in a plan or between actions in a team behavior;
- **coordination** constraints between agents and the actions that they are performing.

This paper describes a method for recognizing defensive plays from spatio-temporal traces of player movement in the

Figure 1: Screenshot of the Rush 2008 football simulator. The offense team (shown in red) is using the play split 8 and being countered by the defense (shown in blue) using a 31 formation (variant 1).

Rush 2008 football game (see Figure 1) and using this information to improve offensive play. To succeed at American football, a team must be able to successfully execute closely-coordinated physical behavior. To achieve this tight physical coordination, teams rely upon a pre-existing playbook of offensive maneuvers [Association, 2000b] to move the ball down the field and defensive strategies [Association, 2000a] to counter the opposing team's attempts to make yardage gains. Rush 2008 simulates a modified version of American football with 8 players per team; plays in Rush are composed of a starting formation and instructions for each player in the formation. These 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 its physical capability.

Although there have been other studies examining the problem of recognizing completed football plays, we present results on recognizing football plays online at an early stage of play. and demonstrate a mechanism for exploiting this knowledge to improve a team's offense. Our system evaluates the competitive advantage of executing a *play switch* based on the potential of other plays to improve the yardage scored and the similarity of the candidate plays to the current play. Our play switch selection mechanism outperforms both the basic offense and a greedy yardage-based switching strategy. Calculating the relative similarity of the current play compared to the proposed play is shown to be a necessary step to reduce confusion on the field and effectively boost performance. .

The remainder of the paper is organized as follows. Section 2 summarizes related work on team behavior recognition in games and simulation environments. Section 3 describes the Rush 2008 football simulator, which was developed from the open source Rush 2005 football game [Rush2005, 2005]. The potential for improving yardage gains through intelligent play choice is discussed in Section 4. Section 5 presents our support-vector based classification approach for early play recognition and results for defensive play recognition. In Section 6 we present our play switching mechanism and introduce the play similarity metric used for calculating switches. We present results for our offensive play improvement procedure (described in Section 7) in Section 8, before concluding the paper with some discussion on the potential uses of early intent recognition.

2 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]. To recognize athletic behaviors, researchers have exploited simple region-based [Intille and Bobick, 1999] or distance-based [Riley and Veloso, 2002] heuristics to build accurate, but domain-specific classifiers. For instance, based on the premise that all behaviors always occur on the same playing field with a known number of entities, it is often possible to divide the playing field into grids or typed regions (e.g., goal, scrimmage line) that can be used to classify player actions. In contrast, we train our classifiers on raw observation traces and do not rely on a field-based marker system.

Most of the camera-based sports analysis work has focused on extracting observation traces, addressing problems such as field rectification and player tracking [Intille and Bobick, 1994]) and has spent relatively little effort on play recognition and opponent modeling. In Intille and Bobick's original system, football play recognition [Intille and Bobick, 1999] is performed on player trajectories using belief networks both to recognize agent actions from visual evidence (e.g., catching a pass) and to determine the temporal relations between actions (e.g. before, after, around). Jug et al. [Jug *et al.*, 2003] used a similar framework for offline basketball game analysis. More recently, Hess et al. demonstrated the use of a pictorial structure model to classify football formations from snapshots [Hess *et al.*, 2007]. These systems were used for post-game analysis of formations and behaviors only and did not address the problem of online intention recognition.

In Robocup, there has been some research on team intent recognition geared towards the Robocup coach competition. 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. This information can be utilized by the coach agent to improve the team's scoring performance. For instance, information about opponent agent home areas can be used triggers for coaching advice and for doing "formation-based marking", in which different team members are assigned to track members of the opposing team. However, the focus of the coaching agents is to improve performance of teams in future games; our system immediately takes action on the recognized play to evaluate possible play switches.

3 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. and the field is 100 yards by 63 yards. The game's objective is to out-score the opponent, where the offense (i.e., the team with possession of the ball), attempts to advance the ball from the scrimmage line into their opponent's end zone. In a full game, the offensive team has four attempts to get a *first down* by moving the ball 10 yards down the field. For this paper we only examine play combinations in isolation, rather than embedded in the full structure of a game. If the ball is intercepted or fumbled, ball possession transfers to the defensive team.

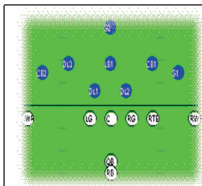	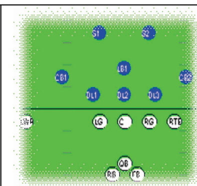	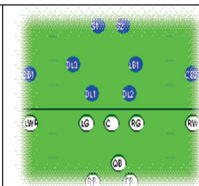
Pro vs 23	Power vs 31	Split vs 2222

Table 1: Three offensive and defensive configurations. Offensive players are shown in white and the defense in blue.

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, directions 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**) 1. 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/full back 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 from the **split** formation is given below:

Table 2: Offensive Plays from *Power* Formation

Play Variant	Description
1	handoff to RB
2	handoff to RB
3	handoff to RB
4	handoff to FB
5	pass towards the left
6	pass using hook routes
7	pass to FB
8	general pass play

- the quarterback will pass to an open receiver;
- the running back and full back 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.

Figure 1 shows an example execution of the above passing play being countered by the the defense using a **31** formation (variant 1). The quarterback has already thrown the ball which is currently in the air between the 40 and 50 yard lines. In Rush defensive plays, the players are given the role of guarding zones of the field or pursuing specific offensive players. In Figure 1, the defense is countering with this allocation of players to tasks:

- the defensive linemen are chasing the quarterback;
- the linebacker is pursuing the running back;
- the cornerbacks are following their respective wide receivers;
- Safety 1 is guarding the high zone;
- Safety 2 is guarding the middle zone;

Table 2 gives general descriptions of possible plays that can be executed from the **power** starting positions using the lineup: QB, RB, FB, WR1, TE1, OL1, O2, OL3. Rush teams have a roster of offensive and defensive players, each possessing unique physical capabilities, which are specified in a game configuration file using a ten point scale to designate the player's power, speed, skill, and endurance. The team compositions are loosely modeled after players on various NFL teams. The experiments described in this paper were run with the Atlanta Falcons (offense) vs. the New England Patriots (defense).

A player's physical capabilities affect his running speed, ability to handle the ball, and to block and tackle other players. In a mechanical sense, Rush treats both players and the ball as 2-dimensional rectangular objects capable of infinite acceleration. As soon as a player or the ball starts to move, it takes on a constant velocity, with the exception that the ball will accelerate downwards due to gravity. When objects overlap, a collision occurs. A collision between players may result in a tackle if a player is carrying the ball or performing a block for the ball carrier.

Table 3: Yardages for Best and Worst Offensive Choices

Offense	**Defense**	**Best (yds)**	**Worst (yds)**	**Avg (yds)**
power	23	10.84	2.41	5.82
power	31	12.82	2.04	5.83
power	2222	7.82	3.67	5.19
power	2231	9.56	3.82	6.93
pro	23	11.34	2.52	6.65
pro	31	17.3	4.77	9.32
pro	2222	23.71	5.47	9.99
pro	2231	16.95	6.78	11.35
split	23	14.74	6.65	10.26
split	31	14.96	3.53	10.71
split	2222	51.82	6.01	17.13
split	2231	51.81	7.43	17.91

4 Competitive Advantage of Intention Recognition

This section discusses how knowledge of the opponents' intended play affects the yardage scored by teams in Rush 2008. Although the players' physical capabilities affect the outcome of individual events, such as blocks, handoffs, and distance covered, the play choice is of key importance in determining the total yardage scored on a play. Each play specifies a particular allocation of players to tasks and positions; different allocations leave various openings in the field which could be exploited by the opponent. Unlike in real football, it is not possible for players to conceal the location of the ball in the Rush simulator to divert attention away from the ball carrier. However, we show that recognizing the opponents' intention can still confer significant competitive advantage by examining the expected outcome of teams executing different play combinations. For the purposes of this paper, we focus on yardage gained over a single play, independent of the team's down or other plays that the team has executed in the past.

To study the effectiveness of different plays, we ran each play combination 50 times (a total of 38400 games) to determine the expected yardage for each play combination in the teams' playbooks and examine the impact of play selection on yardage gained. Table 3 clearly shows that there is a large difference between the best response case for the offense (the offense playing their best play vs. the worst defense), the worst scenario (the offense playing their weakest play vs. the best defense), and the average yardage scored for all play combinations using that pair of formations. Although we don't have any direct control over the defense's choice of formation and play variant, we could in theory increase our yardage by playing our best response play. However, without prior information about the defense's choice of play, the best way to gain this competitive advantage is through accurate early play recognition, gaining knowledge of the other team's intent sufficiently early in the play to exploit holes in the current defense.

5 Play Recognition using SVM

In this paper we focus on intent recognition from the viewpoint of the offense: 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 should 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.

There are 12 initial configurations for the players (choice of 3 formations for the offense and 4 for the defense). We formulate the problem as follows. Let $\mathcal{A} = \{a_1, a_2, \ldots, a_{16}\}$ be the set of agents in the scenario, where the index of each agent maps to its role, as given by the starting configuration. We observe the 2D position of each agent at every time step, enabling us to construct an observation vector, $\mathbf{x}^{(t)}$ at time t that is a concatenation of the observed states of every agent. Thus, the dimensionality of the observation vector is $32t$. Since the offense can select from a set of 8 plays from each initial configuration, the goal of the intent recognition is to output a label $y^{(t)} \in \{1 \ldots 8\}$, and the baseline accuracy for this task is 12.5%.

We perform this classification using support vector machines [Vapnik, 1998]. Support vector machines (SVM) are a supervised binary classification algorithm that 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 the support vector machine projects data points into a higher dimensional space, specified by a kernel function, and computes a maximum-margin hyperplane decision surface that separates 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. When multiple classes receive the highest vote, we select the winning one with the lowest index; the benefit of this approach is that classification is deterministic but it can bias our classification in favor of lower-numbered plays. For a real game system, we would employ a randomized tie-breaking strategy. 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 $t \in \{2, 3, 4, 5, 6, 7, 8, 9, 10\}$ time steps. A multiclass SVM (i.e., a collection of 28 binary SVMs) is trained for each of these cases. 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 paralellizable). Cross-validation 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.

5.1 Evaluation of Play Recognition

For the experiments reported in this paper, we collected a testing set of Rush games with 10 instances of every combination of offensive and defensive plays from each of the different starting configurations. We created observation vectors from this test set for each of the six selected timesteps, resulting in a data set with 72 configurations, each with 640 instances (10 instances of each offense play vs. a defense play). The dimensionality of the problem ranged from 64 (for the shortest observation vector) to 320 (for the longest). Deterministic tie breaking was used to return a forced choice for plays with equal numbers of votes (lowest play number wins).

Table 4 summarizes the experimental results for different lengths of the observation vector (time from start of play), 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 offense plays (perceptual aliasing), however by $t = 3$, the positions of the offensive team are distinctive enough to be reliably recognized. The only case where there is insufficient information to discriminate between play variants is when the defense is using formation 23. Play variant 1 and play variant 2 in this formation are extremely similar, differing only in the deployment of 2 players; hence, play variant 1 is consistently misclassified as being play variant 2 even at $t = 10$.

6 Offensive Play Switches

To improve offensive performance, our system evaluates the competitive advantage of executing a *play switch* based 1) on the potential of other plays to improve the yardage scored and 2) the similarity of the candidate plays to the current play. First, we train a set of SVM models to recognize defensive plays at a particular time horizon as described in the previous section; this training data is then used to identity promising play switches. A play switch is executed:

1. after the defensive play has been identified by the SVM classifier
2. if there is an stronger alternate play based on the yardage history of that play vs. the defense
3. if the candidate play is sufficiently similar to the current play to be feasible for immediate execution.

Rather than calculating play similarity based on executions of individual traces, for every play combination, we create a probability distribution model (shown in Table 5) to describe the players' positions over time, based on the training data. We use this probabilistic representation of the team's spatial-temporal traces to determine the similarity between the plays, using a feature set described in the next section. To determine whether to execute the play switch for a particular combination of plays, the agent considers the set of all offensive plays shown to gain more than a threshold ϵ value. The agent then selects the play in that list most like the current play for each play configuration and caches the preferred play in a lookup table.

When a play is executed, the agent will use all observations up to, and including observation 3 to determine what play the defense is executing before performing a lookup to determine the play switch to make. The process is ended with execution of a change order to all members of the offensive team. Calculating the feasibility of the play switch based on play similarity is a crucial part of improving the team's performance; in the results section, we evaluate our similarity-based play switch mechanism vs. a greedy play switching algorithm that focuses solely on the potential for yardage gained.

6.1 Play Similarity Metric

To calculate play similarities, we create a feature matrix for all offensive formation/play combinations based on the training data. To store the spatio-temporal traces used to calculate play similarity, we create a probability distribution model of the movements l of all offensive players $A = \{a_1, \ldots, a_8\}$ for all time steps t, based on our training samples (s). Let $\sum_s A_l t$ indicate the number of times player A visited location l at $time = t$. The probability athlete A will visit location L at time t is calculated using the formula:

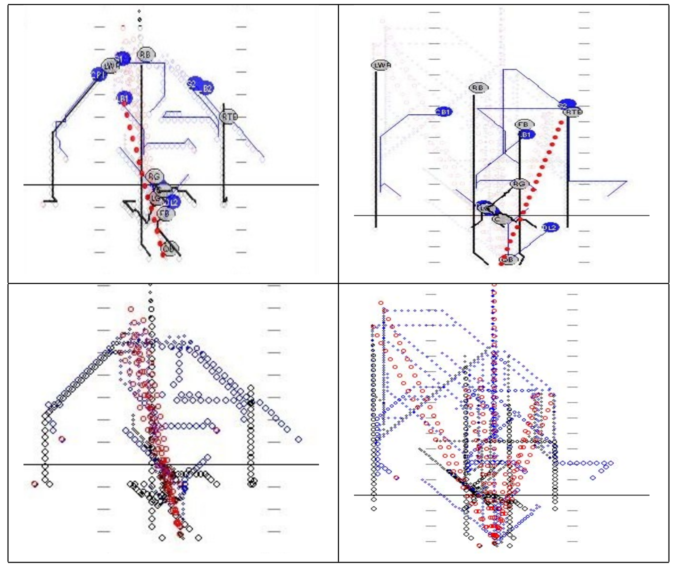

Table 5: Single execution trace from Power-3 vs 2222-3 (top left) and Power-5 vs 2222-1 (top right). The ball position is marked in red. Probabilistic trace models (bottom) for the same plays; increased dot sizes indicate higher probabilities of the player (or ball) being at a particular location at a given time.

$$P(A_{l_t}) = \frac{\sum_s A_l t}{50}$$

For each player and every location the player visits, we store the probability the player will be at a specific location at a given time, and the players four initial movements are used to create a feature vector.

The features collected for each athlete A are

Max(X): The rightmost position traveled to by A

Max(Y): The highest position traveled to by A

Min(X): The leftmost position traveled to by A

Min(Y): The lowest position traveled to by A

Mean(X): $= \frac{\sum_{i=0}^{N-1} X_i}{N}$

Mean(Y): $= \frac{\sum_{i=0}^{N-1} Y_i}{N}$

Median(X): $= Sort(X)_{i/2}$

Median(Y): $= Sort(Y)_{i/2}$

FirstToLastAngle: Angle from starting point $(x1, y1)$, to ending point $(x2, y2)$, is defined as $atan\left(\frac{\triangle y}{\triangle x}\right)$

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

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

Table 4: Play recognition results

Offense	**Defense**	$t = 2$	3	4	5	6	7	8	9	10
Power	23	12.50%	87.50%	87.50%	87.20%	87.28%	87.24%	87.24%	86.94%	86.83%
Pro	23	12.50%	87.50%	87.50%	87.57%	87.24%	87.65%	87.61%	87.83%	87.54%
Split	23	12.50%	87.50%	87.50%	87.39%	87.46%	87.54%	87.87%	87.24%	87.43%
Power	31	12.50%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
Pro	31	12.50%	100.00%	99.96%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
Split	31	12.50%	100.00%	100.00%	100.00%	100.00%	100.00%	99.96%	99.96%	99.96%
Power	2231	12.50%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
Pro	2231	12.50%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
Split	2231	12.51%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	99.96%	99.93%
Power	2222	12.47%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
Pro	2222	12.50%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
Split	2222	12.50%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%

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

Total_Path_Distance: $= \sum_{i=0}^{N-1}\left(\sqrt[2]{{x_i}^2+{y_i}^2}\right)$

Feature set F for a given play c contains all the features for each offensive player in the play and is described as

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

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, but generalized to handle multi-player trajectories. To compare plays, we use the sum of the differences (L_1 norm) between each feature F_{ci}. This information is used to build a similarity matrix M_{ij} for each possible offensive play combination as defined below.

$$M_{ij} = \sum_{c=0}^{\left\|\overrightarrow{F_c}\right\|-1} \Delta\overrightarrow{F_c}$$
$$i, j = 1 \ldots 8$$

There is one matrix M for each offensive formation O_β, where $\beta = \{\text{pro, power, split}\}$ are the offensive formations. Defensive formation/play combinations are indicated by $D_{\alpha p}$, where $\alpha = \{23,\ 31,\ 2222,\ 2231\}$ and p represents plays 1..8. *M* for a specific play configuration is expressed as $O_\beta D_{\alpha p} M_i$, given *i (1…8)* is our current offensive play. The purpose of this algorithm is to find a value *j* (play) most similar to *i* (our current play), with a history (based on earlier observation) of scoring the most yardage. This process is accomplished for every offensive play formation against every defensive play formation and play combination. When the agent is constructing the lookup table and needs to determine the most similar play from a list, given current play i, it calls the method, $min(O_\beta D_{\alpha p} M_i)$ which returns the most similar play.

7 Improving the Offense

Our algorithm for improving Rush offensive play has two main phases, a preprocess stage which yields an 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. As described in Section 5 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:

Step 1: Collect data by running the RUSH 2008 football simulator 50 times for every play combination.

Step 2: Create yardage lookup tables for each play combination. This information alone is insufficient to determine how good a potential play is to transition to. The transition play must resemble our current offensive play or the offensive team will spend too much time retracing steps and perform very poorly.

Step 3: Create a probabilistic trace representation for all 50 training plays with field locations and probabilities of the players being observed in these locations.

Step 4: Create feature matrix for all offensive formation/play combinations using the probabilistic trace representation.

Step 5: Create the final play switch lookup table based on both the yardage information and the play similarity.

To create the play switch lookup table, first the agent extracts a list of offensive plays L given the requirement $yards\,(L_i) > epsilon$ where ϵ is the smallest yardage gained in which the agent does not consider changing 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 (smallest value in the matrix) to our current play *i* using $Min(O_\beta D_{\alpha p} M_i)$ and add it to the lookup file.

During execution, the offense uses the following procedure:

1. At each observation less than 4, collect movement traces for each play.
2. At observation 3, use LIBSVM with the collected movement traces and previously trained SVM models to identify the defensive play.
3. Access the lookup file to find $best(i)$ for our current play i.
4. Send a change order command to the offensive team to change to play $best(i)$.

8 Empirical Evaluation

The algorithm was tested using the RUSH 2008 simulator for ten plays on each possible play configuration in three separate trials. We compared our play switch model (using the yardage threshold $t = 1.95$ as determined by the quadratic fit) to the baseline Rush offense and to a greedy play switch strategy ($t = max$) based solely on the yardage.

Overall, the average performance of the offense went from 2.82 yards per play to 3.65 yards per play ($t = 1.95$) 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 2) shows the yardage gain varies from as much as 100% to as little as 0.1%. Overall, performance is consistently better every configuration tested. In all cases, the new average yardage is over 2.3 yards per game with no weak plays as seen in the baseline, for example, Power vs. 23 (1.4 average yards per play) and Power vs. 2222 (1.3 average yards per play). Results with $t = max$ clearly shows simply changing to the greatest yardage generally results in poor performance from the offense.

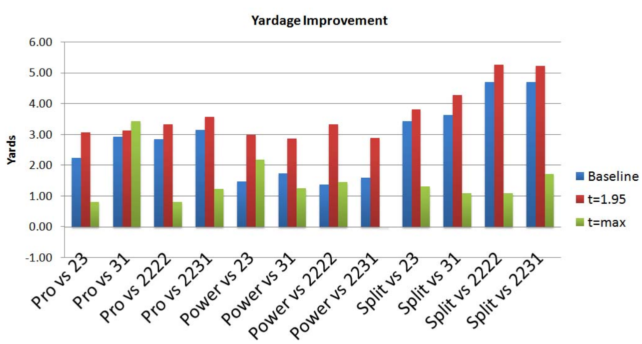

Figure 2: Comparison of play switch selection methods. Our play switch method (shown in red) outperforms both baseline Rush offense (blue) and a greedy play switch metric (green).

Power vs. 23 is dramatically boosted from about 1.5 yards to about 3 yards per play, literally doubling our yardage. Other combinations, such as Split vs 23 and Pro vs. 32 already scored good yardage and improved less dramatically at about .2 to .4 yards more than the gains in the baseline sample. In this table we see all the split configurations do quite well; this is unsurprising given our calculations of the best response. However, when the threshold is not in use and the plays are allowed to change regardless of current yardage, the results are drastically reduced. The reason seems to be associated player miscoordinations accidentally induced by the play switch; by maximizing the play similarity simultaneously, the possibility of miscoordinations is reduced. Figure 3 shows yardage gained by the best play switch strategy over the Rush baseline offense. Power vs 23 experiences the greatest enhancement and Split vs. 31 the least. It is interesting to note Split vs. 23 performed best in the baseline tests and power vs. 23 the worst indicating an inversely proportional expected gain by the algorithm.

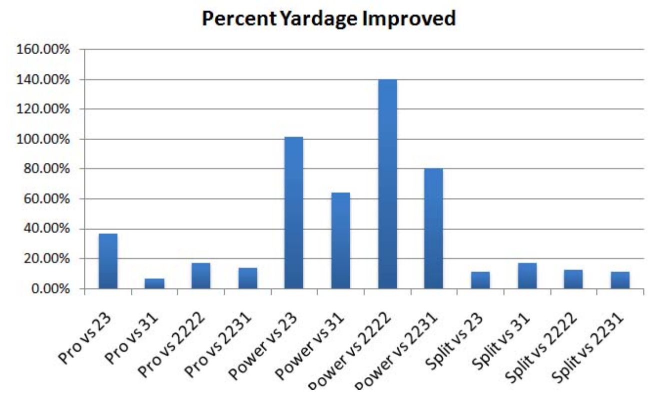

Figure 3: The play-yardage gain over baseline Rush offense yielded by our play switch strategy.

9 Discussion

There are a number of ways that the offensive team could utilize the information provided by an early play recognition system, other than through our play switch mechanism. Another possibility would be to use the play information to predict the future positions of the defensive players and moving towards those coordinates instead of the ones specified in the initial play.

Another question is how to utilize less accurate play recognition to improve offensive play. Since Rush is a simulated game, it is possible for the intent recognizer to perfectly observe the environment, which may be an unrealistic assumption in the real world. To address this, we replicated our experiments under conditions where both training and test data were corrupted by observation noise introduced outside the simulation environment. In this case, classification accuracy at $t = 3$ ranged from 60% to 90%. In this case, it would be possible to have the offensive team consider the information contained in the confusion matrix when evaluating play switches by maintaining multiple hypotheses about the defensive play. The agent could use risk-minimization metrics when considering possible counter responses to multiple hypotheses, choosing to accept smaller, but less risky yardage gains.

Also it is possible that poorer play recognition might not make a difference in all cases if the play can be countered in similar ways. For instance, in formation 23, our SVM classifier persistently misclassifies play variant 1 as 2, resulting in a poor accuracy. However the play only subtly in player action choice, and often not at all in execution trace. In this case, even though the classifier performs poorly, the resulting offense would be unlikely to suffer.

One question is whether the assumption of playbook knowledge, common to any plan recognition system, is reasonable for this domain, since real teams try hard to keep their playbooks secret. A play recognition system would have to compensate for this in the same way that real-life coaches do—by watching and analyzing footage from previous games or from earlier in the same game. We believe that investigating unsupervised and semi-supervised approaches to this problem is a fruitful area for future research; we have demonstrated the applicability of an unsupervised clustering approach for analyzing Rush football plays to improving reinforcement learning [Molineaux *et al.*, 2009].

10 Conclusion

In this paper, we present an approach for early, accurate recognition of defensive plays in the Rush 2008 football simulator. We demonstrate that a multi-class SVM classifier trained on spatio-temporal game traces can enable the offense to correctly anticipate the defense's play by the third time step. Using this information about the defense's intent, our system evaluates the competitive advantage of executing a play switch based on the potential of other plays to improve the yardage scored and the similarity of the candidate plays to the current play. Our play switch selection mechanism outperforms both the basic Rush offense and a greedy yardage-based switching strategy, increasing yardage while avoiding the miscoordinations accidentally induced by the greedy strategy during the transition from the old play to the new one. In future work, we plan to look adapting our current method to be more robust against poor play recognition.

11 Acknowledgments

Gita Sukthankar was partially supported by an ONR Summer Faculty Fellowship (2008).

References

[Association, 2000a] American Football Coaches Association, editor. *Defensive Football Strategies*. Human Kinetics Publishers, 2000.

[Association, 2000b] American Football Coaches Association, editor. *Offensive Football Strategies*. Human Kinetics Publishers, 2000.

[Bhandari *et al.*, 1997] 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.

[Chang and Lin, 2001] 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`.

[Hess *et al.*, 2007] R. Hess, A. Fern, and E. Mortensen. Mixture-of-parts pictorial structures for objects with variable part sets. In *Proceedings of International Conference on Computer Vision*, 2007.

[Intille and Bobick, 1994] S. Intille and A. Bobick. Visual tracking using closed-worlds. Technical Report 294, MIT Media Lab, 1994.

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

[Jug *et al.*, 2003] 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.

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

[Molineaux *et al.*, 2009] Matt Molineaux, David Aha, and Gita Sukthankar. Beating the defense: Using plan recognition to inform learning agents. In *Proceedings of Florida Artifical Intelligence Research Society*, 2009.

[Riley and Veloso, 2000] 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.

[Riley and Veloso, 2002] 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.

[Riley *et al.*, 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.

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

[Rush2005, 2005] Rush2005, 2005. `http://sourceforge.net/projects/rush2005/`.

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

[Wobbrock *et al.*, 2007] 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.