Learning Opponent's Strategies In the RoboCup Small Size League *

Felipe W. Trevizan
fwt@cs.cmu.edu
Machine Learning Department
School of Computer Science

November 27, 2010

Abstract

One of the machine learning challenges posed by the robot soccer domain is to learn the opponents strategies. A team that may be able to do it efficiently may have the advantage to adapt its own strategy as a response to the opponent's strategy. In this work, we propose a *similarity* function to compare two teams, and consequently their strategies, by the ability of one team to mimic the behavior of the other. The proposed function can be used to classify opponents as well as to decompose an unknown opponent as a combination of known opponents. We apply the proposed function to classify opponent's defense strategies in real world data from the RoboCup Small Size League collected during the RoboCup 2007, RoboCup 2008 and USOpen 2009. We also use this *similarity* function to discover patterns in the logs of these championships, such as, similar teams and the number of major defense strategies.

*This work was done under supervision of my advisor Manuela M. Veloso. A part of this work was published in AAMAS 2010 [Trevizan and Veloso, 2010].

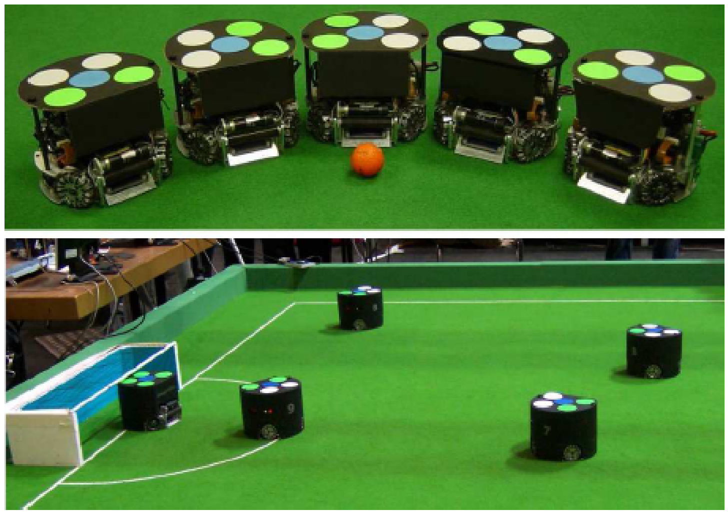

Figure 1: The CMDragons robots (built by Michael Licitra) posed and in a game.

1 Introduction

A multi-agent, dynamic and adversarial domain offers several challenges for machine learning, for instance, learning how the environment evolves and how the adversary behaviors. One example of such domain is robot soccer, in special the RoboCup Small Size League (SSL) [Kitano et al., 1998, Naruse et al., 2010].

The SSL consists of two teams, each one with five robots, that play robot soccer on a field of 6 by 4 meters with global overhead perception and control (Figure 1). Also, the robots must conform to the specifications about their size and shape and they are equipped with kicking devices. SSL robots are typically fast, cruising at speeds of 1-2 m/s while the ball moves at over 4 m/s, and occasionally much faster. The main difference between SSL and the other RoboCup robot soccer leagues are: (i) the allowed use of cameras placed over the field, for shared global perception; and (ii) the allowance of a centralized computer to coordinate the robots, therefore the overall team is autonomous. For this work, we use the logs captured by the CMDragons team [Bruce et al., 2007], the SSL team of Carnegie Mellon University, during 3 championships: RoboCup 2007, RoboCup 2008 and USOpen 2009.

Our approach to classify opponent's strategies uses features extracted from the game logs, such as distance from the ball to the robots and from the CMDragons goal to the robots, and is composed by two steps: (i) segmentation of games into *episodes*; and (ii) the comparison of *episodes*. [1] In the first step, we segment games in a set of small time series called episodes. Each episode encompasses a defense attempted by the opponents and it is obtained by selecting the time intervals in which the game logs registered the employment of an attack strategy by CMDragons. This procedure

[1] In this work we focus in the defense strategies, however all the techniques developed can be directly applied to analyze attack strategies.

assumes that when a team is attacking, the opponent's response is to employ a defense strategy.

In the second step of our approach, we compare two sets of episodes A and B by representing them as matrices E^A and E^B and computing the error of expressing E^A using only a conical combination of the column of E^B. Formally, we compute $d = \min_W \|E^A - E^B W\|_F$ such that $w_{ij} \geq 0$, where $\| \cdot \|_F$ is the Frobenius norm, and define d as a (nonsymmetric) similarity measure from A to B. This measurement can be seem as the ability of team B to mimic the behavior of team A. We can also use the obtained matrix W in the computation of d to explain the behavior of A as a function of B. This is specially interesting when E^A is the set of episodes generated by a new and unknown opponent and E^B is the set of all the episodes seen so far. Then W represents a decomposition of the unknown opponent's strategy as a function of the strategies already known.

The remainder of this paper is organized as follows. In Section 2 we discuss the previous approaches for learning opponent's strategies. In Section 3 we present the data set used in this work. Our approach to learn defense strategies based in the previous games is developed in Sections 4 and 5. In Section 6 we test our proposed approach by presenting a set of experiments, involving classification and pattern discovery. Section 7 brings a few conclusion remarks and future research directions.

2 Related Work

Using logs of the RoboCup Simulation Soccer League, Visser and Weland [2003] tackle a similar problem: classify the behavior of the goalkeeper and the pass behavior of the opponent players. Their approach uses decision tress to label non-overlapping intervals of a given time series. For instance, in the goalkeeper experiments the labels used are: the goalkeeper backs up, the goalkeeper stays in the goal and the goalkeeper leaves the goal. Using the same technique and a different set of labels, they also analyze the pass behavior of the opponent player.

Another work in the simulation soccer is given by Fard et al. [2007]. They proposed an approach to learn opponent's strategy that relies on modeling the opponent as an automaton. One automaton is learned for each previously played opponent by using a predefined payoff matrix, that is designed by an expert, and solving a *Prisoner's Dilemma* game instance. This payoff matrix is defined through high-level features, such as intercept, pass, shoot and dribble and relate the payoff of playing one of this simple strategies with the opponent response (also represented using the same simple strategies). The major difference between this approach and our approach is the usage of domain expert that is consulted for each new team that is considered. Therefore, this approach is not able to automatically analyse an unknown opponent.

For SSL, an alternative approach is given by Bowling et al. [2004] which does not model the opponent's behavior. Instead, their approach to adapt to the opponent is based on the outcome of the attack strategies employed so far. Although this technique has had success when employed in the real games of the SSL, it does not consider previous games against different teams. That is, the authors do not provide a method to relate two teams that play similarly in order to reuse the learned responses.

A more similar approach to the one proposed in this paper is given by Riley and Veloso [2000]. This approach, which is applied to simulation soccer, uses a discretization of the observed features, for instance the position of the robots and the ball, and decision trees to classify opponents. The limitation of this method is the assumption that all opponent's strategies are known a priori.[2] Thus, they focus in the problem of how to classify an employed defense strategy as one of the known strategies. This is the major difference with respect to our approach, since we do not assume any a priori knowledge of the opponent's strategy.

Another work related to SSL and pattern recognition is given by Vail and Veloso [2008]. Instead of classifying opponent's strategies, they focus in the problem of activity recognition. More specifically, a framework using conditional random fields, a temporal probabilistic graphical model, is developed to classify robots by a set of predefined roles, including attacker, marker and defender. This framework could be extend to classify opponent's strategy, however, it would be necessary to provide labeled data to the algorithm. That is, as in [Riley and Veloso, 2000], the opponent's strategies should be known a priori.

Similar to [Vail and Veloso, 2008], Ball and Wyeth [2003] classify the roles of each opponent and instead of using conditional random fields, a naive bayes classifier is applied. Their experiments consist in classifying the roles of the robots of RoboRoos, a SSL team that the authors had access to the ground truth roles. The authors also suggest a method to classify opponent teams by adding a layer to their system that builds a model of the opponent team based on the empirical probability distribution of the roles of each opponent robot. No experiment is provided for team classification. Although in their approach no a priori knowledge about the opponent's strategy is necessary, the proposed method requires labeled data about the robots roles. That is, it is necessary a series of examples of robots playing in the role of a goalkeeper, an attacker, a defender, etc.

3 The Data Set

The data set used in this work is the collection of 13 games played by the CMDragons team during the RoboCup 2007, RoboCup 2008 and USOpen 2009. Each logged game is a multivariate time series in which a new data point encompasses an interval of $\frac{1}{60}$ seconds (about 16 milliseconds). These time series contain 198 features, most of them continuous, such that x and y position, velocity, acceleration and orientation of the robots and the ball and some discrete features, for instance, state of the game and score. All these features are available to both teams, either directly through the referee system, for example the state of the game and score or indirectly through the vision system of each team. Therefore, our data set represents CMDragons perspective of the game, that is, all the indirect measurements are made through the CMDragons vision system, and therefore are subjected to noise and failure [Bruce et al., 2000]. Besides the features available to both teams, the data set also contains the features defining the CMDragons internal state, such as: current team strategy and current role of each robot.

In this work, we focus only in the following 23 features:

- distance from each robot to the ball (10);

[2] In [Riley and Veloso, 2000], the opponent's strategy is called opponent class.

- distance from each robot to their defense goal (10);
- distance from the ball to each goal (2); and
- CMDragons strategy (1).

All features, except from the current CMDragons strategy, are computed features, that is, they are obtained by applying a function to one or more features in the original game logs. For instance, the distance from a robot to the ball is computed by using the Euclidean distance between the position of the robot and the ball. The motivation to use computed features instead the original features is to build a new set of features that is: (i) invariant to flipping the image obtained by the vision system vertically and/or horizontally; and (ii) invariant to the robots id [Stone, 2000]. Property (i) guarantees that the learned patterns are independent of the side in which CMDragons started the game as well as the left and right orientation in the field. The second property is obtained by anonymizing the robots and sorting the distances between robots (CMDragons and opponents) and landmarks. For example, instead of representing the distance between teammate i and the ball, we represent the distance between the ball and the i-th closest teammate to the ball.

To verify if the considered features can represent well the data set, we apply Principal Component Analysis (PCA) to it [Hastie et al., 2005]. Given a set of points $\mathcal{D}$, PCA consists in an orthogonal linear transformation of $\mathcal{D}$ to a new coordinate system such that the i-th coordinate, also known as i-th principal component, is the direction of the i-th greatest variance of $\mathcal{D}$. Another concept associated to PCA is *energy*: the energy of the first k principal components is the sum of the first k largest eigenvalues of $\mathcal{D}$. By normalizing the energy, it is possible to quantitatively asses how well the first k principal components can represent $\mathcal{D}$.

On Table 1 we show the first four principal components of our data set which correspond to approximately 91% of the total energy. For each principal component, we present the 10 largest weight in absolute value, i.e., the 10 original features of the data set that contribute the most for the given principal component. As one may notice, the proposed features are responsible for most of the first four principal components. This gives evidence that it is sufficient to only use the proposed features.

Using this set of features, we want to learn the defense strategy of the opponent. One possible approach is to analyze the time series defined by the games as a whole. However, these time series might contain several realizations of the opponent's defense strategy. Therefore, we segment each time series in non-overlapping *episodes.*

Definition 1 (Episode) *Given a time series T representing a game, an episode is a maximal segment S of T such that on each time step of S: (i) the game is on; (ii) the ball is in the defensive field of the opponent; and (iii) the current strategy being employed by CMDragons is an attack strategy. Also, if the size of S is smaller than s_{min} or size greater than s_{max}, then S discarded.*

The game is on from the moment the game restarted until either a goal is scored, the ball leaves the field or a fault is made by a robot and it is a variable in the game logs. In order to obtain episodes that contains just one realization of an opponent's defense strategy, we need to make one more assumption since we do not have access to the internal state features of the opponents. This assumption, that reassembles a

1st principal comp.		2nd principal comp.		3rd principal comp.		4th principal comp.	
name	weight	name	weight	name	weight	name	weight
$\mathbf{d_4(o,b)}$	0.5002	$\mathbf{d_2(o,g)}$	0.6973	$\mathbf{d_0(o,g)}$	-0.4941	$\mathbf{d_3(o,g)}$	-0.5013
$\mathbf{d_4(o,g)}$	0.5001	$\mathbf{d_2(o,b)}$	0.6971	$\mathbf{d_1(o,g)}$	-0.4938	$\mathbf{d_3(o,b)}$	-0.5005
$\mathbf{d_3(o,b)}$	0.4983	$\mathbf{d_1(o,b)}$	0.0780	$\mathbf{d_0(o,b)}$	-0.4936	$\mathbf{d_4(o,b)}$	0.4988
$\mathbf{d_3(o,g)}$	0.4981	$\mathbf{d_0(o,b)}$	0.0779	$\mathbf{d_1(o,b)}$	-0.4933	$\mathbf{d_4(o,g)}$	0.4985
$\mathbf{d_2(o,g)}$	0.0387	$\mathbf{d_0(o,g)}$	0.0774	$\mathbf{d_2(o,g)}$	0.1096	$p_x(o_3)$	-0.0292
$\mathbf{d_2(o,b)}$	0.0387	$\mathbf{d_1(o,g)}$	0.0773	$\mathbf{d_2(o,b)}$	0.1093	$p_y(o_4)$	0.0025
$p_x(o_0)$	-0.0146	$\mathbf{d_4(o,b)}$	-0.0296	$p_x(o_3)$	-0.0331	$p_x(t_3)$	-0.0018
$p_x(o_3)$	-0.0047	$\mathbf{d_4(o,g)}$	-0.0295	$\mathbf{d_1(t,g)}$	-0.0019	$\mathbf{d_2(o,g)}$	0.0018
$\mathbf{d_1(o,g)}$	0.0045	$\mathbf{d_3(o,b)}$	-0.0260	$\mathbf{d_0(t,g)}$	-0.0019	$\mathbf{d_2(o,b)}$	0.0018
$\mathbf{d_0(o,g)}$	0.0044	$\mathbf{d_3(o,g)}$	-0.0259	$\mathbf{d_4(t,b)}$	0.0018	$p_x(t_1)$	-0.0017

Table 1: The 10 largest weight in absolute value of the first four principal components of our data set including the proposed features. These principal components correspond to approximately 91% of the total energy of the data set. The proposed features are highlighted. For the name of the features, the following pattern was used: o and t represents, respectively, an anonymous opponent and an anonymous teammate, if an index i is used, the it refers to the i-th opponent (or teammate); b represents the ball; g represents the defense goal; $p_x(a)$ represents the position X of a (similarly for $p_y(a)$) and $d_i(a, b)$ represents the distance between a and i-th closest b to a.

zero-sum game assumption, is that the opponent always employs a defense strategy as a response to a CMDragons attack. Therefore, by the definition of episode and this assumption, each episode contains one realization of the opponent's defense strategy. Another implicit assumption in this work is that all information necessary to characterize the defense strategy in an episode e is contained in e. That is, the order of the episodes is irrelevant.

For all experiments in this paper, we use $s_{\min} = 100$ and $s_{\max} = 3000$. Also, we represent each episode by the mean and standard deviation of each one of its features. Therefore if an episode has t timesteps and f features, it will be represented as a point in $\mathbb{R}^{2f}$ instead of a point in $\mathbb{R}^{f \times t}$. This representation simplifies the processes of comparing two episodes since the episodes can have different length (time duration).

Table 2 presents the number of episodes, the average and standard deviation of the episodes length extracted from the game logs against each opponent. CMDragons are also included in the table by using the same definition of episodes and considering CMDragons as the opponent team. Instead of inferring when CMDragons are employing a defense strategy, we use the ground truth that is contained in the game logs. As one may notice, the team BSmart-07 has less episodes than the amount of goals scored by CMDragons, 5 episodes against 10 goals. This is possible because the definition of episodes does not encompass direct kicks from CMDragons' defense field to the opponent's goal and attacks that lasted less than 100 timesteps.

For the remainder of this paper, we denote by f the number of features considered in the classification task and E^A the matrix with the episodes of team A. Therefore if there are m episodes of team A in the game logs, then $A \in \mathbb{R}^{2f \times m}$. We also denote

	Goals scored by CMDragons	Number of episodes	Length of the episodes	
			Avg.	S.Dev.
BSmart-07	10	5	94.40	25.70
Skuba-08	5	9	149.44	137.52
Botnia-07	10	10	125.20	114.93
Kiks-08	10	17	145.76	116.48
WrightEagle-07	10	30	125.60	135.79
EagleKnight-07	9	46	141.28	145.84
PlasmaZ-08	2	79	105.74	104.56
PlasmaZ-07	5	84	95.09	99.71
Fantasia-08	9	96	174.10	174.45
Zjunlict-08	5	97	111.60	96.14
GaTech-09	10	112	224.35	258.13
Zjunlict-07	7	120	126.26	119.33
RoboDragons-07	8	138	184.47	162.83
CMDragons-09	-	53	57.11	36.49
CMDragons-07	-	196	71.29	47.53
CMDragons-08	-	275	82.31	70.39

Table 2: Statistics about the episodes for each team CMDragons played against in the RoboCup 2007, RoboCup 2008 and USOpen 2009. The table is ordered by ascending number of episodes.

by n the total number of episodes in the game logs and $E \in \mathbb{R}^{2f \times n}$ the matrix with all episodes. In the next two sections we explore the definition of episodes to develop a measurement to compare episodes and to find the most relevant episodes in the CMDragons game log.

4 Comparing Defense Strategies

In this section, we develop a measurement to compare two episode matrices. The first question worth notice is if this measure should be symmetric. We illustrate this problem with the following example: consider three teams (A, B and C) and three defense strategies (s_1, s_2 and s_3); the probability $Pr(\text{current strategy is } Y | \text{team} = X)$ is:

	s_1	s_2	s_3
A	1	0	0
B	1/2	1/2	0
C	1/3	1/3	1/3

The team that most resembles the behavior of A is B since in expectation it has more episodes of type s_1 than C and both A and B do not play strategy s_3. On the other hand, C is the best team to mimic B since it plays strategies s_1 and s_2. Thus, the measurement to compare two teams does not necessarily need to be symmetric.

As one may notice, the previous example can be solved by using the KL-divergence, that is, to compute $D(P||Q) = \sum_{x \in \mathcal{X}} P(x) \log \frac{P(x)}{Q(x)}$, where P and Q are probability distributions over $\mathcal{X}$. [3] However, in the problem of learning the opponent's strategies, the set $\mathcal{X}$ is unknown since we do not know all possible strategies.

To overcome this problem, we propose a measurement that compares how well one team can simulate/mimic another one.

Definition 2 *Function $s(\cdot;\cdot)$. Let E^A and E^B be the episode matrices for team A and team B respectively, then $s(A;B) = \min_W \|E^A - E^B W\|$ such that $w_{ij} \geq 0$, where $\|\cdot\|_F$ is the Frobenius norm ($\|E\| = \sqrt{\sum_i \sum_j e_{ij}^2}$).*

The intuition behind the function $s(\cdot,\cdot)$ is that, the smaller $s(A,B)$ is, the better team B can simulate team A. This because, if $s(A,B)$ is small, then the norm of $E^A - E^B W$ is small, therefore it is possible to reconstruct the episode matrix E^A with low error by using conical combinations of the episodes of B (columns of E^B). Also, each column w_i of the matrix W can be seem as an unnormalized probability distribution over the episodes of B. If B plays according to w_i then the Euclidean norm between the episode i of A and the expected episode of B is minimized. The function $s(\cdot,\cdot)$ is non-symmetric and in the previous example it gives the same result: B minimizes $s(A,\cdot)$ and C minimizes $s(B,\cdot)$.

In order to compute the function $s(\cdot,\cdot)$, we cast the proposed minimization problem as a quadratic programming problem. First, notice that $\|E\|_F = \sqrt{tr(E'E)}$ since E is a matrix defined over the reals. Moreover, for our purposes we can use $s(A,B)^2$ instead of $s(A,B)$ since $s(A,B) \geq 0$ for all A and B. Thus we solve the following program

$$
\begin{aligned}
&\min_W \|E^A - E^B\|_F^2 \\
&\quad \equiv \min_W tr((E^A - E^B W)'(E^A - E^B W)) \\
&\quad \equiv \min_W tr(E^{A'} E^A) - 2tr(E^{A'} E^B W) + tr(W' E^B E^B W)
\end{aligned}
$$

which shows that the problem is a quadratic program in W. In all these forms, the only constraints of the programs are $w_{ij} >= 0$.

Using the function $s(\cdot,\cdot)$ to compute the difference between two episode matrices we can classify teams according to the distance to the teams that we already know. Also, we can use this measurement to find patterns in the data set and relations between the teams that we have played before. In section 6 we explore these ideas through a series of experiments and in the next section we extend the usage of $s(\cdot,\cdot)$ to find the main defense strategies in our data set.

5 Main Defense Strategies

In this section we look to all the episode at once, in order to find a small set of the most relevant episodes. In other words, we want to find a matrix $D \in \mathbb{R}^{2f \times k}$

[3]For the KL-divergence, we consider $0 \log \frac{0}{c} = 0$ for $c \geq 0$ and $c \log \frac{0}{c} = +\infty$, for $c > 0$.

where $k \ll n$ such that it is possible to reconstruct E, with low error, by using linear combinations of the columns of D. Clearly, there is a trade off between k and the reconstruction error and in this section such trade off is explored.

Using D, it is possible to decompose each episode as a function of the columns D. The advantage of this new representation is mainly computational: we can compute an approximation of $s(A, B)$ by using the new representation of E^A and E^B and since this new representation is smaller than the original, a speed up can be obtained.

The problem of finding D such that $D \in \mathbb{R}^{2f \times k}$, k ¿ 0, is equivalent to find a rank k approximation of the matrix E. This can be found by solving a similar optimization problem as the one presented in Section 5: $\min_{B,W} \|E - DW\|F$ such that rank(D) $= k$.

This problem can be solved optimally through the singular value decomposition of E (SVD decomposition). The SVD decomposition of a matrix $M \in \mathbb{R}^{m \times n}$ is the product $U\Sigma V'$, where $U \in \mathbb{R}^{m \times m}$ is unitary, $\Sigma \in \mathbb{R}^{m \times n}$ has nonnegative real numbers on the diagonal and zeros otherwise, and $V \in \mathbb{R}^{n \times n}$ is unitary. [4] This factorization always exists for matrices defined over the reals and $Y = U\Sigma^* V'$, where Σ^* has only the first (largest) k values of Σ, is the rank k matrix that minimizes $\|M - Y\|_F$ [Horn and Johnson, 1990]. In our case, D equals the first k columns of U and W is the first k rows of $\Sigma^* V'$.

Besides the mathematical interpretation of D obtained using the SVD decomposition, this approach does not offer an interpretation in the robot soccer domain since each b_{ij} can be negative. This implies that features whose meaning requires a positive value, such as the standard deviation the closest robot to the ball, can have negative values in this new representation.

In order to get a direct interpretation in our domain, we can enforce that D is composed by subset of the columns of E, i.e., D is a submatrix of E. The hardness of this new problem, referred in the literature as column-based low-rank matrix approximation [Drineas et al., 2006] and CX-decomposition [Hyvönen et al., 2008], is unknown [Drineas et al., 2006]. The best approximation algorithm for this problem is proposed by [Drineas et al., 2006]: given k, ϵ and δ, it finds D and W such that:

$$\|E - DW\|_F \leq (1 + \epsilon)\|E - \tilde{E}_k\|_F$$

with probability at $1 - \delta$ where D has $O(\frac{k^2 \log(\frac{1}{\delta})}{\epsilon^2})$ columns of E and $\tilde{E}_k$ is the best rank-k approximation of E (with no constraint).

In the next section we explore these two decompositions to estimate how many defense strategies exist in our data set. This estimation is also used to speed up the classification task in the experiments presented in the next section.

6 Experiments

We perform 5 experiments to evaluate our proposed similarity function $s(\cdot, \cdot)$. In the first experiment we use the $s(\cdot, \cdot)$ to find similarities between teams and in the second experiment we estimate the number of strategies contained in our game logs. Experiments 3 and 4 evaluate the accuracy of classifying teams according the proposed

[4] We assume that the values $\Sigma_{ii}, i \in \{1, \ldots, \min\{m, n\}\}$, are in descending order.

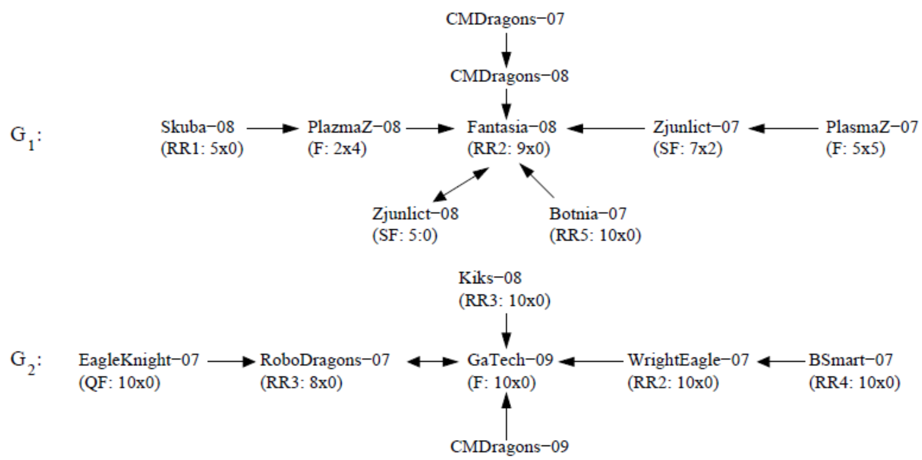

Figure 2: Relations obtained between teams by using the proposed similarity function $s(\cdot,\cdot)$. In the graph, the solid arrows $A \rightarrow B$ represent that $B = \text{argmin}_x\, s(A,x)$ and can be interpreted as "B is team that can better simulate A". Underneath each team name and in parentheses is the score of the game between CMDragons and the given team. It is also depicted the phase of the championship in which the game was played: RR stands for round-robin, QF for quarter-finals, SF for semi-finals and F for final.

similarity function. The last experiment explores the approximated computation of $s(\cdot,\cdot)$ proposed in Section 5.

6.1 Experiment 1: Relation between teams

In the first experiment, we relate the teams that played against CMDragons as well as CMDragons by finding, for each team A, $\text{argmin}_B\, s(A,B)$. The result of this experiment is depicted in Figure 2, where each solid arrow $A \rightarrow B$ represents that $B = \text{argmin}_x\, s(A,x)$. As shown in Figure 2, we obtain a disconnected graph composed by two directed graphs, G_1 and G_2. By looking at the score of each game, one may hypothesize that the teams are separated by their defense strength. That is, teams in G_1 have a better defense than teams in G_2, since CMDragons scored less goals on the teams in G_1 than in the teams in G_2. In order to verify if this hypothesis is true, in Table 3 we present statistics about the games played by CMDragons against each of the teams.

	Goals Scored by CMDragons	Ratio Attack2	Ratio Attack3	Average time to score	S. Dev. time to score	Ratio of the episodes outcome				
						Defense	Direct Free Kick		Indirect Free Kick	
							ours	theirs	ours	theirs
Botnia-07	10	0.9000	0.1000	328.00	209.74	0.3333	0	0	0.6667	0
BSmart-07	10	1.0000	0	509.40	593.51	0	0.5000	0	0.5000	0
EagleKnight-07	9	0.9333	0.0667	1324.44	1273.86	0.1143	0.1143	0.2000	0.2000	0.3714
Kiks-08	10	1.0000	0	1769.30	1323.97	0.1429	0.0714	0.1429	0.3571	0.2857
WrightEagle-07	10	0.9615	0.0385	1999.62	1968.81	0.3500	0.3500	0.3000	0	0
GaTech-09	10	0.9906	0.0094	2634.89	1915.21	0.2927	0.2073	0.1463	0.1341	0.2195
RoboDragons-07	8	0.8168	0.1832	3057.75	2390.57	0.2368	0.0351	0.1754	0.0789	0.4737
Fantasia-08	9	0.9888	0.0112	3916.78	3905.96	0.5238	0.0357	0.0595	0.1429	0.2381
Zjunclict-07	7	0.6552	0.3448	5185.29	5222.76	0.5893	0.0982	0.0804	0.0804	0.1518
PlasmaZ-07	5	0.7286	0.2714	5664.40	4819.29	0.5938	0.0625	0.1719	0.0781	0.0938
Zjunclict-08	5	1.0000	0	7134.40	8105.80	0.6585	0.1098	0.1585	0.0244	0.0488
Skuba-08	5	1.0000	0	14102.33	20114.26	0.8750	0	0.1250	0	0
PlasmaZ-08	2	1.0000	0	14784.50	8610.44	0.6119	0.0299	0.1045	0.0896	0.1642

Table 3: Statistics about the games played between CMDragons and opponent teams. Columns 3 and 4 represent ratio between the two different attack strategies employed by CMDragons; columns 5 and 6 presents statistics about the time to CMDragons score a goal; columns 7 to 11 represent ratio between the possible outcome of an episode. This outcome can be: (7) defense, i.e., the opponent successfully neutralized the attack and started a counter-attack; (8,9) a directed free kick for CMDragons and the opponent; and (10,11) a indirected free kick for CMDragons and the opponent. This table is sorted by ascending average time to score a goal.

Sorting the teams by ascending average time to score a goal (ATSG), we obtain that G_2 contains the teams with the second to the seventh smaller ATSG and G_1 contains the team with smallest ATSG and the teams with the eighth and higher ATSG. This shows a strong evidence that the hypothesis is true, i.e., we clustered the teams between strong defense and regular defense with the exception of only one team. Also, the statistics in Table 3 corroborates with most of the relations obtained, for instance: in 7 out of the 11 relations, the ratio between the two different attack strategies employed by CMDragons is no more than 0.10 different; Skuba-08 and PlasmaZ-08 have the two largest ATSG, PlasmaZ-07 and Zjunlict-07 have the fourth and fifth largest ATSG.

6.2 Experiment 2: Estimating the number of defense strategies

In the second experiment, we estimate how many defense strategies are in our data set. To perform this, we use the Bayesian information criterion (BIC) [Hastie et al., 2005]. The BIC criterion is defined as: given a penalty $\lambda > 0$ and the matrices $D \in \mathbb{R}^{2f \times k}$ and $W \in \mathbb{R}^{k \times n}$, $\text{BIC}(\lambda, D, W) = \|E - DW\|_F^2 + \lambda f k \log n$. This value is the score of the model, and we want to find a model with small score, since it is composed by the error of approximating E by D and W plus a regularization term that penalizes large models, i.e., large values of k. Therefore, for a fixed λ we can vary the value of k and find the one that minimizes $\text{BIC}(\lambda, D, W)$.

Figure 3 presents, for different values of λ, the value k that minimizes $\text{BIC}(\lambda, D, W)$ for the two approaches presented in Section 5, namely, SVD-decomposition and CX-decomposition, to find the matrix D and W given E and k. This plot gives evidence that there are between 7 to 17 different defense strategies in our data set since this range of values lies in the transition phase between the under-constraint (small λ) and over-constraint (large λ) value of BIC.

6.3 Experiment 3: Classification according to the defense strategy

The third experiment consists of classifying the teams by their defense strategy. For a given percentage p, we randomly select p episodes of each team i, denoting these episodes as T^i, and use them to classify the remainder episodes. The remainder episodes R are grouped by team, such that each set of episodes R^i has only samples of the team i. We classify R^i as team j if $T^j = \text{argmin}_x \, s(R^i, x)$.

Besides the classification accuracy, we also present a second measurement called *rank*. The rank of a team i is the position of T^i when all matrices T^j are sorted, in ascending order, by the value of $s(R^i, T^j)$. Therefore if the rank of i is 1, then R^i is correctly classified, since T^i has the minimum value of $s(R^i, T^j)$ for all T^j. Table 4 presents the result of 25 executions of this experiment for p equals 30%, 40% and 50%.

This experiment shows that we can perfectly classify 7 out of 16 of the teams, namely Fantasia-08, Zjunlict-08, GaTech09, Zjunlict-07, RoboDragons-07, CMDragons-07 and CMDragons-08, using 40% and 50% of the data as training. Also, the average rank for EagleKnight-07, CMDragons-09, PlasmaZ-08 and PlasmaZ-07 is at most

	30% for training				40% for training				50% for training			
	Accuracy		Rank		Accuracy		Rank		Accuracy		Rank	
	Avg	S.Dev	Avg	S.Dev	Avg	S.Dev	Avg	S.Dev	Avg	S.Dev	Avg	S.Dev
BSmart-07	0.04	0.20	7.60	4.40	0	0	9.12	3.58	0.12	0.33	7.36	5.43
Skuba-08	0	0	7.28	1.92	0	0	6.80	2.34	0	0	6.92	1.97
Botnia-07	0.20	0.40	6.92	5.13	0.12	0.33	7.40	5.55	0.32	0.47	5.32	4.67
Kiks-08	0	0	6.12	2.58	0	0	5.52	2.97	0	0	5.32	2.64
WrightEagle-07	0	0	5.40	2.94	0.08	0.27	5.12	2.68	0.24	0.43	4.24	2.69
EagleKnight-07	0.64	0.48	2.08	1.86	0.96	0.20	1.16	0.80	0.76	0.43	1.48	1.12
PlasmaZ-08	0.60	0.50	1.56	0.76	0.32	0.47	1.88	0.72	0.44	0.50	1.72	0.73
PlasmaZ-07	0.68	0.47	1.32	0.47	0.44	0.50	1.68	0.74	0.60	0.50	1.48	0.65
Fantasia-08	0.96	0.20	1.04	0.20	1.00	0	1.00	0	1.00	0	1.00	0
Zjunlict-08	1.00	0	1.00	0	1.00	0	1.00	0	1.00	0	1.00	0
GaTech-09	1.00	0	1.00	0	1.00	0	1.00	0	1.00	0	1.00	0
Zjunlict-07	1.00	0	1.00	0	1.00	0	1.00	0	1.00	0	1.00	0
RoboDragons-07	1.00	0	1.00	0	1.00	0	1.00	0	1.00	0	1.00	0
CMDragons-09	0.40	0.50	2.00	1.11	0.28	0.45	2.16	1.06	0.32	0.47	1.96	0.97
CMDragons-07	1.00	0	1.00	0	1.00	0	1.00	0	1.00	0	1.00	0
CMDragons-08	1.00	0	1.00	0	1.00	0	1.00	0	1.00	0	1.00	0

Table 4: Statistics about the classification experiment. For each percentage of the data used for training, we classified each team in the remainder of the data (testing data). This table presents the average and standard deviation of the classification accuracy and the average and standard deviation of the rank for 25 executions of this experiment. The rank of a team A is defined as the position of $s(A', A)$ in the sorted vector of $s(A', \cdot)$, where A' are the instances of A in the test data. This table is sorted by the number of episodes of each team (see Table 1).

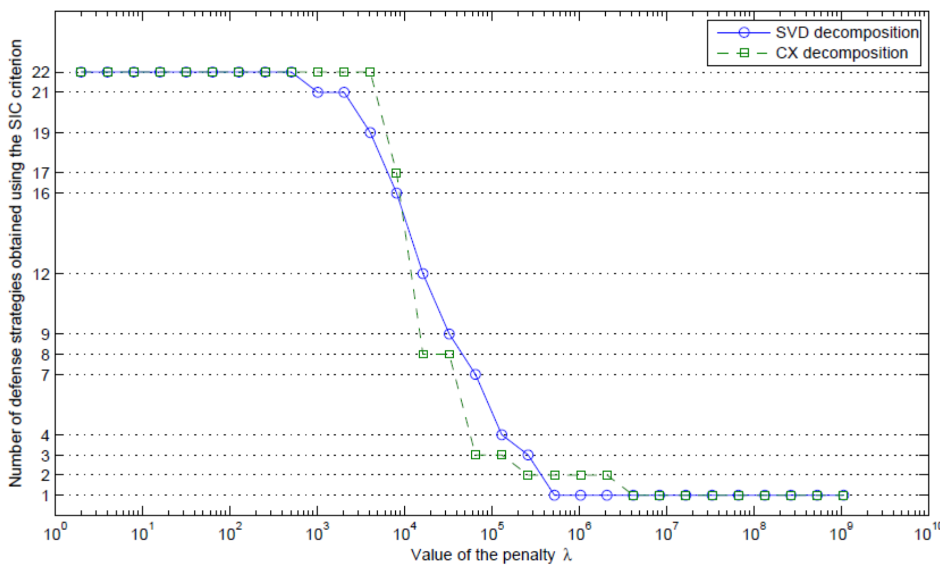

Figure 3: Value k that minimizes the BIC regularization criterion for different values of λ.

2, i.e., the proposed measurement $s(\cdot, \cdot)$ ranked the correct answer as the second most similar team. For the remaining 5 teams, namely Skuba-08, Kiks-08, Botnia-07, WrightEagle-07 and BSmart07, the classification accuracy is not satisfactory since the average rank for these teams is at least 4. One explanation to this poor performance for these teams is that they are the 5 teams with the least amount of episodes in our data set (Table 2).

6.4 Experiment 4: Classification of random mixture of teams

The fourth experiment is an extension of the previous one. Instead of using teams R^i with only samples of team i for testing, a random mixture of the teams $\hat{R}^i$ is used. The testing set $\hat{R}^i$ is built by selecting 15 random episodes of team i and one episode of each team $j \neq i$. Therefore, the probability of an episode in $\hat{R}^i$ is from team j is $\frac{1}{2}$ if $i = j$ and $\frac{1}{30}$ otherwise. Given the values of $s(\hat{R}^i, \cdot)$, we define the induced probability distribution $\hat{P}(\hat{R}^i$ is the team $j)$ as proportional to $\frac{1}{s(\hat{R}^i, T^j)}$, where T^j is the matrix with the training episode for team j. Table 5 contains the results of this experiment. The KL-divergence between the original distribution and the obtained $\hat{P}$ is smaller, i.e. differ less, than the uniform distribution (random guess) in half of the cases. By looking at the mode of $\hat{P}$, i.e. $\operatorname{argmax}_x \hat{P}(\hat{R}^i$ is the team $x)$, we observed that for all teams, except CMDragons08, the mode of $\hat{P}$ is either GaTech-09 or Fantasia-08. This is interesting since these two teams are the centers of each graph in Figure 2.

	KL-divergence	Mode	Pr(Mode)	Pr(real Mode)
GaTech-09	0.3753	GaTech-09	0.1625	0.1625
Fantasia-08	0.4174	Fantasia-08	0.1457	0.1457
RoboDragons-07	0.4745	GaTech-09	0.1369	0.1313
CMDragons-08	0.4791	CMDragons-08	0.1382	0.1382
CMDragons-07	0.5342	GaTech-09	0.1341	0.1208
Zjunlict-08	0.6400	Fantasia-08	0.1370	0.0921
Zjunlict-07	0.6711	Fantasia-08	0.1257	0.0887
CMDragons-09	0.7012	GaTech-09	0.1468	0.0822
Kiks-08	0.7502	GaTech-09	0.1636	0.0722
PlasmaZ-08	0.8111	Fantasia-08	0.1387	0.0675
WrightEagle-07	0.8418	Fantasia-08	0.1316	0.0583
PlasmaZ-07	0.8560	Fantasia-08	0.1379	0.0587
Skuba-08	0.8900	Fantasia-08	0.1505	0.0591
EagleKnight-07	1.0590	GaTech-09	0.1155	0.0367
BSmart-07	1.1018	GaTech-09	0.1215	0.0315
Botnia-07	1.1302	Fantasia-08	0.1410	0.0306

Table 5: Results of the classification experiment using random mixture of the teams. For each team A, it was select 15 random episodes of A and one episode of each other team. The remaining of the data set is used for training. The first column contains the KL-divergence between the original distribution and the one obtained by our proposed method. The KL-divergence between the original distribution and the uniform distribution is 0.7254, therefore, our method performs better than the random guess for the 8 teams (top 8 lines). The second column contains the mode of the induced probability distribution, i.e., the team that has maximum probability and the third column its probability; and the fourth column presents the induced probability of the mode of the original distribution (the team on each line). The induced probability distribution was obtained by averaging 25 runs of the experiment. This table is sorted by ascending KL-divergence.

	Class. Accuracy		Rank		Running time (secs)	
	Avg.	S.Dev.	Avg.	S.Dev.	Avg.	S.Dev.
Exact computation	0.53	0.50	3.49	4.21	6.09	3.86
SVD decomp., k=17	0.52	0.50	3.69	4.20	5.82	4.11
CX-decomp., k=17	0.33	0.47	5.14	4.63	4.81	3.74
SVD decomp., k=12	0.51	0.50	3.68	4.17	5.04	4.80
CX-decomp., k=12	0.35	0.48	4.85	4.42	4.61	3.47
SVD decomp., k=7	0.46	0.50	3.77	4.17	4.44	6.36
CX-decomp., k=7	0.34	0.48	4.78	4.38	3.67	5.20

Table 6: Statistics about classification, rank and running time using the exact computation of $s(\cdot,\cdot)$, the SVD decomposition approximation and the CX-decomposition approximation. The statistics were obtained by 25 runs of the experiment. The settings of this experiment is the same as the third experiment (Table 4) using 50% of the data for training. To make the table easier to read, the result for different team were combined.

6.5 Experiment 5: Approximating the value of $s(\cdot;\cdot)$

In the last experiment we compare the approach presented so far, i.e., the exact computation of $s(\cdot,\cdot)$, with the approximated approach suggested in Section 5: to use a rank k approximation of the episodes matrix to decrease the number of features describing each episode. For this experiment, 50% of the data set was used as training set and remaining 50% as testing set. The chosen values of k, namely 17, 12 and 7, are based in the second experiment (Figure 3). The results are presented in Table 6 and as expected, the classification accuracy, the rank and the running time decrease as k decreases. One may also notice that the exact computation and the SVD decomposition for $k = 17$ approaches achieve almost the same classification accuracy and rank, however the SVD approach is about 0.2 seconds faster than the exact computation approach.

7 Conclusions and Future Work

In this paper we have introduced a novel approach to compare team strategies. This approach relies on the best approximation, according to the Frobenius norm, of the matrices representing the episodes in our data set of each team. Therefore, we consider that a team A is similar to a team B, if the episode matrix of A is best approximated by a conical combination of the episodes in the episode matrix of B.

We presented experiments, using real data from the RoboCup 2007, RoboCup 2008 and USOpen 2009, showing how classification can be performed using the proposed measurement. We also applied this measurement to find similarities in the defense strategies of the teams in our data set. The obtained patterns are corroborated by the presented statistics of the games played by CMDragons against these teams.

Possible future research directions include extending the proposed approach to handle episodes represented as time series instead of the representation by mean and standard deviation used in this work. This extension is non-trivial since each episode has different lengths (time duration). The trivial extension of applying the same definition of $s(\cdot,\cdot)$ in episode tensors, i.e., matrices in $\mathbb{R}^{f\times t\times n}$ where t the length of the episode, does not work, thus additional research is needed to find a suitable approach.

A second general direction for further investigation is to explore adaption according to the opponent. That is, to use of the knowledge from the previous opponents when playing against an unknown opponent through the proposed decomposition of the unknown strategies.

Acknowledgments

The authors thank the past and current CMDragons teams members, in particular Stefan Zickler, Joydeep Biswas and James Bruce, for developing and sharing the CMDragons log data used in this work.

References

[Ball and Wyeth, 2003] Ball, D. and Wyeth, G. (2003). Classifying an Opponents Behaviour in Robot Soccer. In *Proceedings of the 2003 Australasian Conference on Robotics and Automation (ACRA).*

[Bowling et al., 2004] Bowling, M., Browning, B., and Veloso, M. (2004). Plays as effective multiagent plans enabling opponent-adaptive play selection. In *Proceedings of International Conference on Automated Planning and Scheduling (ICAPS04)*, volume 1.

[Bruce et al., 2000] Bruce, J., Balch, T., and Veloso, M. (2000). Fast and inexpensive color image segmentation for interactive robots. In *Proceedings of the IEEE/RSJ Internation Conference on Intelligent Robots and Systems – IROS'00*, volume 3, pages 2061–2066, Japan. IEEE Computer Society Press.

[Bruce et al., 2007] Bruce, J., Zickler, S., Licitra, M., and Veloso, M. (2007). Cmdragons 2007 team description. Technical Report CMU-CS-07-173, Carnegie Mellon University, School of Computer Science.

[Drineas et al., 2006] Drineas, P., Mahoney, M., and Muthukrishnan, S. (2006). Subspace sampling and relative-error matrix approximation: Column-based methods. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 316–326.

[Fard et al., 2007] Fard, A., Salmani, V., Naghibzadeh, M., Nejad, S., and Ahmadi, H. (2007). Game Theory-based Data Mining Technique for Strategy Making of a Soccer Simulation Coach Agent. In *6th International Conference on Information Systems Technology and its Applications,(ISTA'07)*, pages 54–65.

[Hastie et al., 2005] Hastie, T., Tibshirani, R., Friedman, J., and Franklin, J. (2005). The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85.

[Horn and Johnson, 1990] Horn, R. and Johnson, C. (1990). *Matrix analysis*. Cambridge Univ Press.

[Hyvönen et al., 2008] Hyvönen, S., Miettinen, P., and Terzi, E. (2008). Interpretable nonnegative matrix decompositions. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 345–353. ACM.

[Kitano et al., 1998] Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., Osawai, E., and Matsubara, H. (1998). Robocup: A challenge problem for ai and robotics. *RoboCup-97: Robot Soccer World Cup I*, pages 1–19.

[Naruse et al., 2010] Naruse, T., Laue, T., and Weitzenfeld., A. (2010). Small size robot league. http://small-size.informatik.uni-bremen.de/.

[Riley and Veloso, 2000] Riley, P. and Veloso, M. (2000). On behavior classification in adversarial environments. *Distributed autonomous robotic systems*, 4:371–380.

[Stone, 2000] Stone, P. (2000). *Layered Learning in Multiagent Systems: A Winning Approach to Robotic Soccer*. MIT Press.

[Trevizan and Veloso, 2010] Trevizan, F. W. and Veloso, M. M. (2010). Learning Opponent's Strategies In the RoboCup Small Size League. In *9th International Conference on Autonomous Agents and Multi-Agent Systems*. Springer.

[Vail and Veloso, 2008] Vail, D. and Veloso, M. (2008). Feature selection for activity recognition in multi-robot domains. In *Proceedings of AAAI*.

[Visser and Weland, 2003] Visser, U. and Weland, H. (2003). Using online learning to analyze the opponents behavior. In *RoboCup 2002: Robot Soccer World Cup VI*, pages 78–93. Springer.