Footprint based Retrieval for a CBR system employed in a RoboSoccer domain

FIRA 2015
Team Sahas

S R Manikandasriram (EE11B127)
Surajkumar Harikumar (EE11B075)

Department of Electrical Engineering, IIT Madras

09 May 2015

Outline

Outline

Problem Statement

CBR for defensive robosoccer agent

- Consider only 1 opponent attacker, 2 defenders and our goalie
- Action selection for coordinated defense behaviors
  - block the goal, steal the ball - no parameters
  - clear, tackle the opponent - parameters
- Episodic knowledge for improving rule based approach

The focus is on analysing *Footprint cases* using the concept of Competance Groups.

Our Contribution

- Finding a derived feature space which covers the problem space and has intuitive and relevant features
- Defined the notion of *Solves* which is crucial towards using Footprint based Retrieval
- Acquired case data of features and actions for defenders and goalie through rules designed by domain experts
- Developed a CBR engine for decision making
- Developed tools for visualizing and analysing data
- Analysed the relative coverage of cases based on the adopted similarity measures

Outline

Outline

State space $\rightarrow$ Feature space

- State Space: observations from simulator (pos,vel,ang) for each robot and the ball
- Feature Space: transformed space which also covers the whole problem/state space.
- Features are relevant for decision making using the CBR system

Decision making is non-linearly related to state space. Transformation to feature space helps remove non-linearity, and simplify similarity measure definitions

Features Used

- Ball position and velocity w.r.t goal
- Position of two nearest defenders and attacker w.r.t ball
- Angle of ball w.r.t bots direction
- goalie position w.r.t ball

16 features in total

Output of the CBR system

The output is described as an action for each robot.

- Actions are abstract tasks
- The game state is used to decide how to go about an action.
- There are 2 kinds of actions
  - Non-Parameterized - Uses only game state data to execute action e.g Block ypos, spin clear
  - Parameterized - Uses additional information, provided by CBR system e.g Block line, intercept the ball

Outline

Similarity Measures

We use a difference / ratio distance function, and an approximate sigmoid function defined as

$$f(d) = \frac{1}{2}\left(1 - \frac{k}{1+|k|}\right); \quad k = \frac{d-\theta}{\alpha} \tag{1}$$

where θ is the point of similarity 0.5, and α gives the rate of decay of the sigmoid.
We use a preference order and a weighted sum for global similarity.

Retrival and Adaptation

- Retrieval - We use priority queue to get closest k cases
- Adaptation - Since the output is an abstract action which uses the problem's game state to execute action, there is little need for adaptation.
- For parameterized actions, we can use weighted sum of parameters of retrieved cases.

Outline

Solves

Def 1: RetrievalSpace(t) $= \{c \in C$: c is retrieved for t$\}$

- The RetrievalSpace of a case is controlled by the Similarity Measures.

Def 2: AdaptationSpace(t) $= \{c \in C$: c can be adapted for t$\}$

Def 3: Solves(c,t) iff $c \in$ [RetrievalSpace(t) $\cap$ AdaptationSpace(t)]

Coverage

Def 4: CoverageSet(c) = {c' ∈ C:Solves(c,c')}

Def 5: ReachabilitySet(c)={ c' ∈ C:Solves(c',c)}

Def 6: RelativeCoverage(c) =

$$\sum_{c' \in CoverageSet(c)} \frac{1}{|ReachabilitySet(c')|}$$

Coverage

RelatedSet(c) = CoverageSet(c) $\cup$ ReachabilitySet(c)
SharedCoverage(c1,c2) iff RelatedSet(c1) $\cap$ RelatedSet(c2)

- We can then define a Competence Group (maximal sets of independent cases with shared coverage)
- In every competence group, we can find the cases with highest relative coverage that solve all cases in group. This gives the footprint set.

Computation

- RetrievalSpace is governed by retrieved k cases for every case
- AdaptationSpace is determined if the solution actually solves the problem
- A directional graph is generated based on solves
- The in degrees and out degrees of a node give the Coverage and Reachability Set sizes

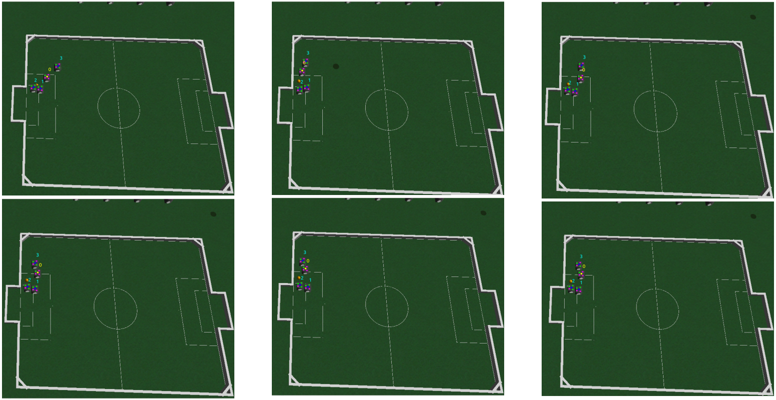
Outline

Data collection

A sample strategy used for collecting data from the system

- **Goalie**
  - If *OurBall* and *goalie_is_closer*, then *goalie_clear_ball* else *goalie_defend_ball_ypos*
  - if *not OurBall* and *on_goal*, then *goalie_defend_ball_extrapolated_position*, else *goalie_defend_ball_ypos*
- **Defenders**
  - If *OurSide* and *not OurBall*, then *defender_steal* else *defender_pass_clear*
  - If *not OurSide*, then *defender_sine*
- **Attacker**
  - If *TheirSide*, then *move_ball_to_goal*

Data Visualization



Training

There are two aspects to training the system:

- Random restarts from different initial configurations
- Running different strategies for attacker and defender
  - If the different strategies produce contradicting behaviours, the conflicts need to be resolved

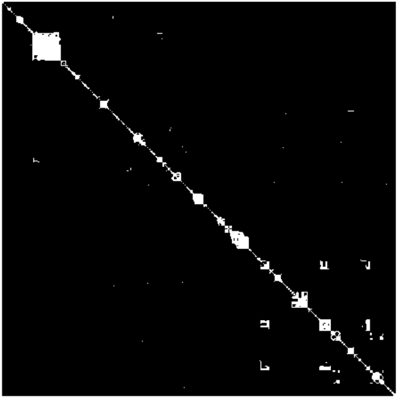
Outline

Case Base

- We ran 2 strategies and sampled every 0.5 seconds to gather 663 cases.
- These were chosen to be non conflicting in action

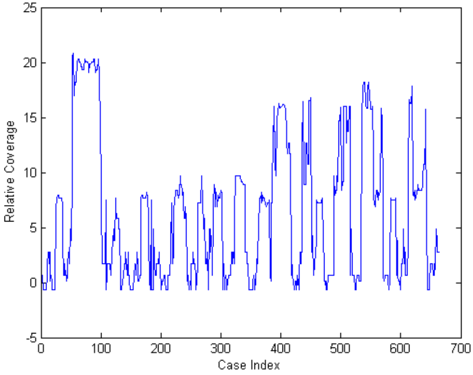
Case Base as Image

Using the notion of *Solves*, we generated a graph of which cases help solve others. The image for the graph is shown below



Relative Coverage

We then computed the relative coverage and shared coverage for each case.



Outline

Inferences from Heatmap

- Sparse matrix. Only 8055 edges in a 663×663 graph
- Solid Squares along diagonal. Similar cases are adjacent in the temporal space
- 4 corner squares. These correspond to recurrence of a scenario at different time instances
- Can be used to analyse casebase alignment using Case Base as an Image Metaphor
- High relative coverage corresponds to the squares in the heatmap

Ack: Case base as an Image Metaphor, Sutanu et. al

Limitations

- Dataset is biased to single strategy. Aggregating data from multiple strategies would give new insights
- Few behaviours (i.e. small solution space). Adding more specialized actions would let the game evolve differently
- Adding adaptation knowledge will help in discovering new competance groups

Outline

Conclusion

- Implemented CBR Engine for a defensive AI agent. Defined feature space, similarity measures, behaviours (inherently capturing adaptation).
- Data collection using rule based approaches for imparting human domain expert knowledge
- Used notion of *Solves* to obtain a heatmap which gives useful insights
- Used relative coverage to detect competence groups
- Analysed the major limitations of our work and pointed out various possible extensions

Questions

Questions?