Predicting Situated Behaviour Using Sequences Of Abstract Spatial Relations

Jay Young and Nick Hawes
University of Birmingham, United Kingdom

Abstract

The ability to understand behaviour is a crucial skill for Artificial Intelligence systems that are expected to interact with external agents such as humans or other AI systems. Such systems might be expected to operate in co-operative or team-based scenarios, such as domestic robots capable of helping out with household jobs, or disaster relief robots expected to collaborate and lend assistance to others. Conversely, they may also be required to hinder the activities of malicious agents in adversarial scenarios. In this paper we address the problem of modelling agent behaviour in domains expressed in continuous, quantitative space by applying qualitative, relational spatial abstraction techniques. We employ three common techniques for Qualitative Spatial Reasoning – the Region Connection Calculus, the Qualitative Trajectory Calculus and the Star calculus. We then supply an algorithm based on analysis of Mutual Information that allows us to find the set of abstract, spatial relationships that provide high degrees of information about an agent's future behaviour. We employ the RoboCup soccer simulator as a base for movement-based tasks of our own design and compare the predictions of our system against those of systems utilising solely metric representations. Results show that use of a spatial abstraction-based representation, along with feature selection mechanisms, allows us to outperform metric representations on the same tasks.

Introduction

In co-operative or adversarial tasks, knowledge of an opponent or one's own team-mates might reveal common mistakes in strategy that can be exploited, or identify areas where the performance of a team can be improved. The usefulness of such models, should they be properly exploited, is clear. However, we must first tackle the issues surrounding how Artificial Intelligence systems can be built to learn robust, predictive models of behaviour from often complex and chaotic multi-agent domains.

Quantitative representations, while being rich in information, do not lend themselves well to generalisation, and can be brittle when used to encode models of varying situated behaviour. There may also be forms of richness in the domain that lie as-yet undiscovered that first require some interpretation of the metric data before being revealed. In this study, we are interested in investigating the *abstract spatial relationships* that exist between entities, and how they can be used as a tool to construct predictive models of behaviour.

Our approach build on a broad body of recent work in the field of Qualitative Spatial-Relational Reasoning, which seeks to provide tools for human-like, common-sense reasoning about space, time and motion by providing discretisations over quantitative measurements. These techniques may also be *relational*, and in our work we argue that in adversarial and/or co-operative domains, agent behaviour can be explained in relational terms.

To ground our study in human experience, we consider the example of a human football player unable to verbally communicate with his or her team-mates (Stone, Kaminka, and Kraus 2010). While all parties may have an understanding of the rules of the game, the particular team strategy and tactics employed are not known to our player until they are observed. We might imagine that while there may exist no explicit verbal communication facilitating co-ordination, the human player in this scenario would communicate with their team-mates through modification of the environment (changing the position of the ball, and themselves), and would make behavioural decisions based on spatial information regarding the positioning, motion and orientation of friendly and opposing players. In such situations human reasoning is typically conceptual, abstracting away from metric values, allowing for generalisation and the composition of robust, transferable models.

RoboCup Soccer

Simulated Robocup soccer is a widely used platform for Artificial Intelligence research wherein teams of AI agents compete in games of virtual soccer. It is a fully-distributed, adversarial multi-agent domain, and therefore the ability to predict opponent behaviour – both in terms of low-level tactics or high-level strategy - can directly improve team performance. Simulated agents are subject to sensor and action models, which introduces a stochastic element.

Background

A variety of techniques in the existing work have been applied to player modelling in the RoboCup domain, such as traditional reinforcement learning (Vieira, Adeodato, and Gon 2010; Jung and Polani ; Shimada, Takahashi, and Asada 2010; Molineaux, Aha, and Sukthankar 2008), and case-based reasoning (Ros, Llu, and Mantaras ; Floyd, Esfandiari, and Lam 2008). Little work investigates the application of qualitative, relational representations in regard to the problem. Though (Bombini et al. 2010) attempted to characterise player behaviour using relational sequence models, this was performed using pre-defined relational symbols over the action space such as dribble, shoot and pass. Molineaux, Aha, and Sukthankar employ a similar abstraction, though consider just co-operative behaviour acquisition.

Recent work (Sridhar and Cohn 2010) has used QR techniques in the recognition of events in an aircraft loading domain, representing the interactions between a trolley, an aircraft, a plane puller and a loading bridge in terms of a qualitative, relational representation. Changing sequences of relations over time then form the basis of predictive models of events in the domain. We argue that to model behaviour in complex, continuous domains, we can make use of similar calculi. However, we are interested particularly in studying how several qualitative calculi can be applied concurrently, all derived from the same metric data, to provide AI systems with more rich spatial information. For instance, in a real game of football we might consider the situation of a player attempting to shoot the ball into a goal which is defended by a goalkeeper. When shooting at the goal, the relative angles between the player, the goalkeeper and the goal posts are important. The goalkeeper would like to minimise the angle between himself and a goalpost with respect to the player, so as to make the shot as difficult as possible, whereas the player would prefer to widen the angle to make the shot easier. This angle will not be precisely specified for either the player or the goal keeper, and in fact there may be a range of angles that are all satisfactory.

Our Approach

In our approach, we assume a learning agent that possesses a birds-eye view of the simulated football stadium. The agent observes the ongoing game in real-time, and receives low-level metric data informing it of the position and orientation of the game ball, all players on the pitch, and the actions they take. Systems with this design have been studied with regard to the automatic generation and understanding of narratives of football games, towards systems capable of automatic commentary generation (Hajishirzi et al. 2012a; 2012b). On each discrete time step of the simulation, our agent observes the quantitative state of the world and abstracts it into sets of qualitative relations according to the implemented calculi (described in the following section) to create a state representation. This state representation is then used as the basis for a predictive model, which we discuss and evaluate in more detail later.

Qualitative Relational Representations

Abstraction has been characterised as a non-injective mapping over the elements of a set S to a target space T (Frommberger and Wolter 2010), compressing a range of possible inputs into a target space. In AI the field of Qualitative Reasoning (QR) looks at how such abstract models can be used. We are particularly interested in abstractions that are also *relational*, and one of the most well-known examples of such is Allen's interval algebra (Allen 1983), which allows qualitative reasoning about temporal relations between events, such as “Event X occurs before event Y” and “Event X overlaps with event Y” among others. Qualitative symbols can then explain a wide range of quantitative observations an AI system might make, aiding knowledge transfer and generalisation, and improving the scalability of learning systems (Frommberger 2010).

In our work we are interested in spatial abstractions applied to entities moving in physically-analogous space. We are interested in how we can employ spatial abstractions to discretise over the attributes mobile entities typically possess, such as position, orientation and velocity, and replace them with symbols expressed in the form of binary relations. For now we constrain ourselves to the use of just three separate calculi, specified in some of their simplest forms, and of which we now provide an overview.

Region Connection Calculus

Introduced in the early 90s (Randell, Cui, and Cohn 1992) RCC is used to reason about relations between spatial regions. Between regions a set of binary, boolean relations is formed from the primitive *Connected(x,y)* relation, which holds when given regions share common points in space. Additional relations can then be formulated by considering the *degree* of connectivity between regions. In our work we employ regions described by simple rectangular shapes, however RCC can also be used to reason about arbitrary shapes and concave regions (Ouyang, Fu, and Liu 2007). Most recently, RCC has been applied to diverse areas such as activity recognition (Sridhar and Cohn 2010), image analysis (Falomir, Jim, and Escrig 2011), and spatial reasoning for mobile robots (Hawes, Klenk, and Lockwood 2012).

The set of relations we employ in our work is known as RCC5 and provides the following binary relations.

- Equal (**EQ**) - Regions share all points in common.
- Disconnected (**DC**) - No points are shared between regions.
- Overlapping (**O**) - Some points are shared, however there also exist points that are not shared.
- Proper Part (**PP**) - One region entirely encloses the other.
- Proper Part Inverse (**PPi**) - One region is entirely enclosed by the other.

We encode the underlying structure of the environment as a set of rectangular regions which mirror the spatial configuration of a football pitch, representing areas such as the left and right halves, penalty areas, the centre circle, goal areas, and so on. This is based on information provided by the RoboCup simulator. The structure of a standard field is shown in *Figure 1*. RCC relations between these regions are calculated in an exhaustively pairwise manner, with only one relation holding between a given pair at any one time. From this, we can formulate a symbolic representation of the pitch structure in terms of RCC5 relations – for instance, considering the centre circle on the pitch, we can say that it *Overlaps* both the left and right halves of the pitch, and is *Disconnected* from the penalty areas. We can also say that the left

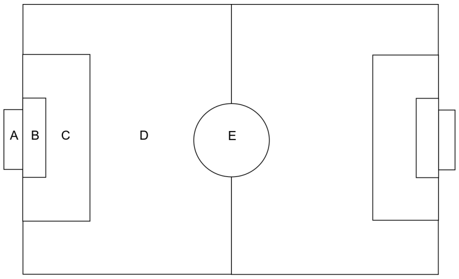

Figure 1: Standard football pitch structure. The areas labelled are as follows. A is the left inner goal area. B is the left outer goal area. C is the left penalty area. D is the left half, and E is the centre circle. Similar region labels hold for the right side of the pitch.

and right penalty areas and goal areas are *Proper Parts* of their respective pitch halves.

We also represent in-game players and the game ball as mobile regions, employing minimum bounding rectangles around the entities. The regions used to represent non-static entities are party to the same relations described previously, and their relationships are similarly calculated in an exhaustively pairwise manner. Transitions between relations are governed by a conceptual neighbourhood, which ensures that relations cannot transition directly should intermediate relations exist between the two which must first be observed (Gooday and Cohn 1994). That is, in every relational state, there exists a local neighbourhood of valid transitions.

Qualitative Trajectory Calculus

QTC is used to represent qualitative symbols describing information about moving objects (Van de Weghe et al. 2005), and has seen recent applications in reasoning about motion in robotics (Hanheide, Peters, and Bellotto 2012) and traffic management situations (Delafontaine 2011).

The QTC calculus itself has several variants – QTC$_N$ allows for reasoning about the movement of points in a network, and QTC$_S$ allows for reasoning about the *shapes* of trajectories of moving points. For this study, we employ the QTC$_B$ (Basic) relation set, which encodes the following relations between any two points *k* and *l*, with each relation taking on a value in the domain {-,+,0} describing its state.

- Towards/Away relation

  - : *k* is moving away from *l*
  + : *k* is moving towards *l*
  0 : *k* is stable with respect to *l*

- Left/Right relation

  - : *k* is moving to the left of *l*
  + : *k* is moving to the right of *l*
  0 : *k* is stable with respect to *l*

- Relative Motion relation

  - : *k* is moving faster than *l*
  + : *k* is moving slower than *l*
  0 : *k* is stable with respect to *l*

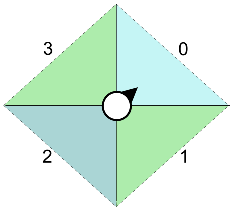

Figure 2: A set of four angular zones egocentric with respect to a player, such that the 0^{th} zone is always aligned with the orientation of the player's body.

Transitions between QTC relations are governed by a conceptual neighbourhood (Van de Weghe and Maeyer 2005). Our technical implementation follows that of (Delafontaine, Cohn, and Van de Weghe 2011), using the position, velocity and orientation of points to generate relations.

We decompose each multivariate relation into a set of boolean relations corresponding to each of the values in its domain, giving us nine QTC relations in total (with only one relation per group of three active at any one time). Doing this allows us to represent RCC5, QTC (and later Star calculus) relations in a common format. We apply the QTC calculus to all *moving* entities, specifically players and the game ball.

Star Calculus

The Star calculus is a calculi for describing qualitative direction between points in space with respect to one-another, and is done so in terms of a set of binary relations (Renz and Mitra 2004). Star employs angular zoning based on either an adjustable granularity parameter m, in which case the uniform angular division between zones is simply $360/m$, or by specifying a custom, arbitrary sequence of angles. The result is a set of circle sectors emanating from a point, extending into infinity, discretising over a range of angles, with each composing a single binary relation. Between two points then, the current Star relation is determined by taking the angle between them and determining which discrete zone the result falls in to.

The Star calculus has been employed in cognitive robotics as a component of knowledge-representation systems (Daoutis, Coradeschi, and Loutfi 2012), as a tool for generating qualitative models of route networks, and in localisation (Renz and Wölfl 2010) and navigation tasks (Stolzenburg 2010).

In our work, the Star relations that hold between all mobile entities are calculated on each time step along with RCC and QTC relations. We also ensure that Star relations are considered in egocentric terms, relative to the frame of reference of an entity. A rough example of this is illustrated in *Figure 2*. For players, we orient the relation set such that the 0^{th} relation is always the relation directly in front of the player's body.

Representing States

We have described our implementation of the RCC5, QTC$_B$ and Star calculi, with the underlying commonality being that all calculi encode information as sets of binary, boolean relations. In our system, we unify these relations into a single representation which describes the state of the environment, in terms of the relations that hold between all mobile entities as well as their relationships with the pitch structure. The current state is then given as a vector of boolean values expressed over the possible set of all relations.

Relational representations are prone to state-space explosion, and ours is no exception. To partly address this, we engage in a trivial preprocessing step to remove redundant relations by removing those that we know *a priori* will remain static, such as the relations between regions of the pitch structure, and relations between entities and themselves. These things provide us no descriptive power, and so we discard them. There may still exist other redundant relations in our representation, but this is not something we can be aware of without encoding domain-specific information, which we wish to avoid. We initially deliberately over-generate the representation, and only later on, after analysing the data from trials do we begin pruning the state space.

Learning

Our aim is to produce a model that will provide us the ability to predict the *actions* a player we are interested in is likely to perform. In RoboCup Soccer, players select between a finite set of possible actions to execute on each frame, of which we consider a simplified sub-set as follows.

- Turn - Allows a player to alter their orientation.
- Dash - Causes the player to move in the current direction.
- Kick - Kicks the ball with some degree of force.

We also consider that an agent might not choose to perform any of these actions on a given frame, and so we allow for a NOOP action. Other actions exist, such as those for catching the ball and tackling, however we do not consider them for now, as the tasks *what* we set do not make use of them. In addition, there exist actions for altering field of view or communicating with other players. We do not consider these either, as they do not directly affect the spatial relations that exist between players (though they may do, depending on the design of an agent). Several actions also have parameters (for instance, how hard to kick a ball), however for now we are only interested in predicting the action *label*. That is, we are only concerned about predicting what type of action is to be performed, and we do not think about the quantitative component associated with it.

The approach we take is to utilise a Hidden Markov Model (HMM), wherein our latent states are the agent actions we wish to predict, and observations are given in the form of vectors of qualitative world states. Given the current time t, and a window of length w of previous observations (qualitative world states) where $y(n)$ is the world state at some previous point in time n where $n < t$, our goal is to determine a probability distribution over the latent variable $x(t)$ – the action to be taken by player x. That is, we look for $P(x(t)|y(t-w),..,y(t))$, which we determine using the Forward algorithm (Rabiner and Juang 1986). This gives us a distribution over actions to be taken by a player given a history of previous evidence in the form of a sequence of states, of which we take the action with the max probability as our prediction. However, as discussed previously, our system considers a *global* state space. This has pitfalls – in our case, an agent cannot act on something it cannot perceive (or has not perceived recently in some finite time horizon), and so attempting to predict based on a global state will mean that activities that occur beyond the agent's perception act as a form of noise. Before building the HMM, we engage in a pre-processing step to filter out irrelevant information, and to determine which state variables provide us the most predictive power.

Feature Selection

Before we employ the HMM to predict agent actions, we must first determine what variables in the state space may be relevant to performing that prediction. We do this in two steps, first we prune variables according to a filter based on the agent's field of view, and then we analyse those observed variables to see which provide the most information about an agent's behaviour.

Perception Filtering

We filter the state representation at each step by including only those relations that are perceivable by the agent whose actions we wish to predict. For each time step, we start with the global state S as a vector of state variables of binary relations $x \in S$ and an agent a, and generate a perception filter by applying a weighting function over the state variables to give us $S‘ = w(a,x)x \; \forall x \in S$. The weighting function $w(a,x)$ returns 1 iff the entities represented in the relation x are perceivable by the agent a, and 0 otherwise. We make use of the egocentric Star relations, illustrated in *Figure 2*, to implement this. A RoboCup agent has a field of view of 90° which we break down into three 30° zones (which we label, $s0$, $s1$ and $s11$. Front, front-right and front-left respectively), and in order for an agent to perceive a relation, the entities involved in that relation must fall into one of these zones. That is, $w(a,x)$ returns 1 iff one of these relations holds.

Mutual Information

Our second step in the feature selection process seeks to Given the set of abstract spatial relations observed by an agent, we want to find those that appear to provide the most information about what the agent will do next. That is, what spatio-temporal changes influence the agent's behaviour – what are its *relevant inputs*.

We do this partly for practical purposes, but also to maintain a degree of generality in our representation. Calculating relations exhaustively pairwise creates a state-space explosion, but this is initially desirable, as it allows our representation to maintain somewhat domain agnostic at this stage – it may be the case that some relations provided by our QR calculi have no use in a particular application domain, however we do not know this ahead of time. Instead, we prefer to generate all possible relations, and determine which ones are relevant in a pre-processing step.

There may be a wide range of ways to achieve this, but we employ the information-theoretic measure of *Mutual Information* (MI), allowing us to quantify levels of mutual dependence between variables (Guyon 2003). MI is given in its simplest form as

$$I(X,Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \frac{p(x,y)}{p(x)(p(y)}$$

MI can be thought of as the reduction in entropy in Y given X. We may imagine in our case that X represents the sequence of states perceived by an agent where $x_t \in X$ is the vector of state variables at time t, and $y_t \in Y$ represents the action taken in a state. So, we wish to discover which state variables provide good entropy-reduction with regard to the action performed.

To do this, we employ an algorithm loosely based on that of (Station and Guyon 2004), which utilises the idea of *Conditional Mutual Information*. Conditional MI, that is $I(X,Y|Z)$, allows us to determine the reduction in entropy in Y given that we know *both* X and Z.

Our MI-based filtering algorithm is as follows. We begin with $S = \{\}$ an empty set and O as the time-series of qualitative, spatial relations, where O_t is the particular state vector observable by by an agent a at time t, and the action taken in that state $y_t \in Y$. We iterate over O, and for each entry select a variable $z \in O_t$, and calculate the conditional mutual information score $CMI(S,Y|z)$. That is, the information provided about Y, given that the state representation contains the observable relation z in addition to the variables already added to S. This means that we look for features that provide good mutual information in conjunction with every other feature in the set. If the information score is high (In our case – better than guessing the values of the variables, and higher than some threshold parameter we may wish to set) we include the variable in S. We initialise the algorithm by forward search, adding to S the first variable that provides MI with Y. From there, we start considering the CMI between the remaining state variables and the contents of S. This allows us to identify variables that are weakly dependent, and produces a state representation populated with variables that, when observed, are informative regarding the agent's next action. The feature selection algorithm also culls those variables which remain static, and provide no predictive power. The new, filtered state representation is then used to provide the basis for estimated probability distributions used as HMM parameters.

Evaluation

We split our experiments down into three tasks with each task involving RoboCup soccer agents of our own design attempting to accomplish some goal related to navigation and interaction with the environment.

Task 1 - Unobstructed Passing

In our first task, illustrated in *Figure 3*, the agent labelled A is our subject of interest. The agent's task is to pass the ball, which is placed in front of it, to the friendly agent labelled D. D patrols the goal area (movement paths are shown as dashed lines), whereas adversary agents B and C patrol in roughly concentric circles around A – the movement model that each agent is subject to ensures a stochastic component, so the patterns are not exact. A will not pass the ball if its shot is blocked by B or C, or if D is not located within the illustrated cone. Once A has a clear shot, it will pass the ball to D and the simulation will reset, with B, C and D being placed on random points on their movement paths. This constitutes one training example. In our experiment, our training set contains 208 examples of this scenario being played out. In this task, at each time step the agent may choose to kick the ball or to do nothing.

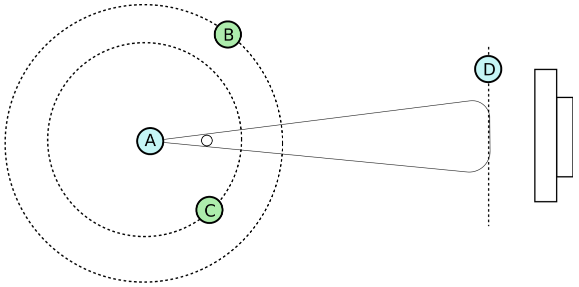

Figure 3: Illustration of environment for first task

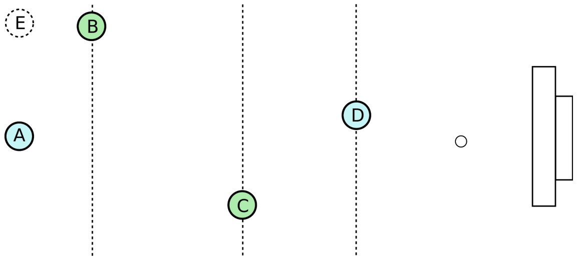

Figure 4: Illustration of environment used for second and third tasks

Task 2 - Gauntlet

In this scenario, the agent A must move forwards from its starting position to the ball, which it must then kick towards the goal. Blocking its movement are the agents B, C and D. When A's movement is unobstructed (no agent occupies a cone identical to the one in *Task 1*), it will move forwards towards the ball, and will stay in place otherwise. This requires that the agent use both dash and kick actions, but it may also decide to do nothing in a particular state. Our training set for this task contains 205 examples.

Task 3 - Altered Gauntlet

The set-up for this task is the same as in the second task, however the agent now starts at the position marked E on the diagram. This alters the situation, such that after completing the gauntlet task, the agent must then turn to face the ball before moving towards it. This then makes use of all three actions – dash, turn and kick – as well as the option to do nothing in a particular state. Our training set for this task contains 201 examples.

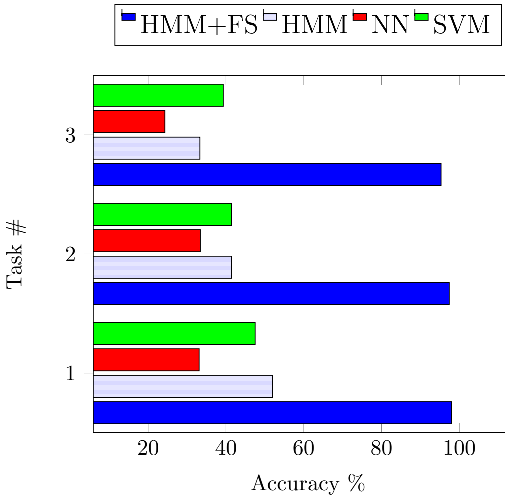

Figure 5: Performance of systems on tasks using QR representation.

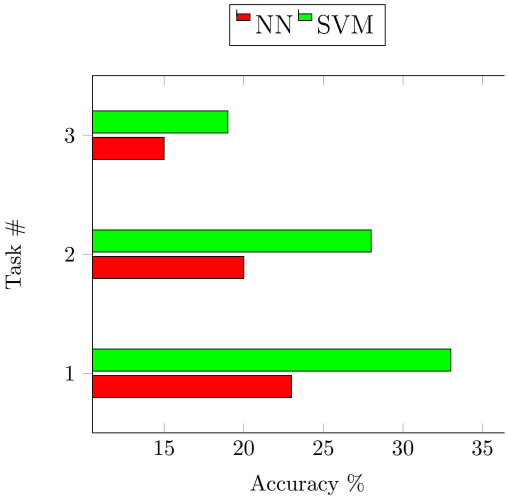

Figure 6: Performance of systems on tasks using metric representation.

Results

Figure 5 shows our results comparing four different approaches to learning. All utilise our qualitative state representation. First, a HMM+FS which implements the full feature selection algorithm, with both perception filtering and MI filtering. Second, the HMM with no feature selection except for the perception filter. Third, a standard feed-forward Neural Network trained with resilient backpropogation, and finally a Support Vector Machine. All prediction systems employ a sliding window of 5 previous frames. The NN and SVM do not implement the full feature selection algorithm, though do implement the perception filter. *Figure 6* shows results for the NN and SVM on the same tasks, however this time utilising a metric-level representation as opposed to our QR representation. No feature selection or filtering is performed on this data.

We see that results are poor without feature selection. This is because our feature selection mechanisms – perception and information-based filtering – reduce the number of variables in the state representation to a fraction of the initial number. This, in turn, means that we deal with a smaller number of unique states which are then repeated often. Without this effect, and considering only the global state, we see a much larger number of unique states (though still fewer than if we had used a metric representation) since this means that there is a far higher number of unique variable combinations to consider. This hinders the estimation of probability distributions used to construct HMM parameters, since, in short, if a state is seen only once in the training set, it provides little predictive power.

In Task 1, the full, global representation initially contains 975 state variables. After feature selection, the set contains just 86 variables, producing 19 unique states observed by the agent (compared to 795 unique, global states before feature selection). The variables with the highest MI scores are often those representing Star relations between agent A and the other agents in the scene. In one particular state, the variables with the highest MI scores are $ml(A:B), mr(A:C), s0(A:D)$ – the first is the "moving to left" QTC$_b$ relation, the second is the "moving to right" QTC$_b$ relation, and the third is the Star relation corresponding to an angular zone of $30\,^{\circ}$ egocentric with A's orientation. If these variables are all true, the state is one in which the agents B and C are not blocking A's shot, and D is within the required angular range. We see very similar results for the other tasks, with information about orientation and motion providing the highest MI scores. Analysis of these scores reveals that there is some degree of overlap between calculi – specifically QTC$_b$ and Star, which often represent the same form of information. Attempting to find such overlaps could be one form of improvement to our system in the future. In addition, MI tells us how important a variable is in relation to others in a set, however we simply use this as a way to produce an incusion-exclusion filter over the set of state variables. Once accepted, all variables have the same degree of importance in the state representation, and we discard the measure of relative importance uncovered by the MI analysis.

Conclusion

We presented an approach to predicting the behaviour of agents based on utilising several different qualitative, spatial abstraction calculi in unison. We were able to first explode, and then prune a state-space of abstract spatial relations utilising a preprocessing step based on measures of mutual information. This allowed us to locate relations that were most relevant to predicting an agent's behaviour. We now seek to apply our system to real-world data, in the form of tournament-level RoboCup soccer games.

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement No 600623 and the EPSRC grant EP/K014293/1.

References

Allen, J. F. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM* 26(11):832–843.

Bombini, G.; Mauro, N. D.; Ferilli, S.; and Esposito, F. 2010. Classifying Agent Behaviour through Relational Sequential Patterns. *Proceedings of the 4th KES international conference on Agent and multi-agent systems* 273–282.

Daoutis, M.; Coradeschi, S.; and Loutfi, A. 2012. Cooperative knowledge based perceptual anchoring. *International Journal on Artificial Intelligence Tools* 21(3).

Delafontaine, M.; Cohn, A. G.; and Van de Weghe, N. 2011. Implementing a qualitative calculus to analyse moving point objects. *Expert Systems with Applications* 38(5):5187–5196.

Delafontaine, M. 2011. *Modelling and analysing moving objects and travelling subjects: Bridging theory and practice*. Ph.D. Dissertation, Ghent University.

Falomir, Z.; Jim, E.; and Escrig, M. T. 2011. Describing Images Using Qualitative Models and Description Logics. *Spatial Cognition & Computation* 11(1):1–31.

Floyd, M.; Esfandiari, B.; and Lam, K. 2008. A case-based reasoning approach to imitating RoboCup players. *21st International Florida Artificial Intelligence Research Society Conference*.

Frommberger, L., and Wolter, D. 2010. Structural knowledge transfer by spatial abstraction for reinforcement learning agents. *Adaptive Behavior* 18(6):507–525.

Frommberger, L. 2010. *Qualitative Spatial Abstraction in Reinforcement Learning*. Ph.D. Dissertation, University of Bremen.

Gooday, J., and Cohn, A. 1994. Conceptual neighbourhoods in temporal and spatial reasoning. *Spatial and Temporal Reasoning, ECAI*.

Guyon, I. 2003. An Introduction to Variable and Feature Selection 1 Introduction. 3:1157–1182.

Hajishirzi, H.; Hockenmaier, J.; Mueller, E. T.; and Amir, E. 2012a. Reasoning about RoboCup Soccer Narratives. *arXiv preprint arXiv:1202.3728.*

Hajishirzi, H.; Rastegari, M.; Farhadi, A.; and Hodgins, J. K. 2012b. Semantic Understanding of Professional Soccer Commentaries. *arXiv preprint arXiv:1210.4854*.

Hanheide, M.; Peters, A.; and Bellotto, N. 2012. Analysis of human-robot spatial behaviour applying a qualitative trajectory calculus. *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication* 689–694.

Hawes, N.; Klenk, M.; and Lockwood, K. 2012. Towards a Cognitive System That Can Recognize Spatial Regions Based on Context. *Proceedings of the 26th National Conference on Artificial Intelligence*.

Jung, T., and Polani, D. Learning RoboCup-Keepaway with Kernels. *JMLR: Workshop and Conference Proceedings* 1:33–57.

Molineaux, M.; Aha, D. W.; and Sukthankar, G. 2008. Beating the Defense : Using Plan Recognition to Inform Learning Agents.

Ouyang, J.; Fu, Q.; and Liu, D. 2007. A Model for Representing Topological Relations Between Simple Concave Regions. *ICCS* 160–167.

Rabiner, L. R., and Juang, B. H. 1986. An introduction to hidden Markov models. *ASSP Magazine,* 3:4–16.

Randell, D.; Cui, Z.; and Cohn, A. 1992. A spatial logic based on regions and connection. *3rd International Conference on Knowledge Representation and Reasoning*.

Renz, J., and Mitra, D. 2004. Qualitative direction calculi with arbitrary granularity. *Lecture notes in computer science*.

Renz, J., and Wölfl, S. 2010. A qualitative representation of route networks. *Proceedings of the 2010 conference on ECAI* 1091–1092.

Ros, R.; Llu, J.; and Mantaras, R. L. D. A Case-Based Approach for Coordinated Action Selection in Robot Soccer. (February 2009):1–50.

Shimada, K.; Takahashi, Y.; and Asada, M. 2010. Efficient Behavior Learning by Utilizing Estimated State Value of Self and Teammates. *Robot Soccer World Cup XIII* 1–11.

Sridhar, M., and Cohn, A. 2010. Unsupervised learning of event classes from video. *AAAI* 1631–1638.

Station, C., and Guyon, I. 2004. Fast Binary Feature Selection with Conditional Mutual Information. 5:1531–1555.

Stolzenburg, F. 2010. Localization, exploration, and navigation based on qualitative angle information. *Spatial Cognition & Computation* 28–52.

Stone, P.; Kaminka, G.; and Kraus, S. 2010. Ad hoc autonomous agent teams: Collaboration without pre-coordination. *AAAI'10.*

Van de Weghe, N., and Maeyer, P. D. 2005. Conceptual neighbourhood diagrams for representing moving objects. *Perspectives in Conceptual Modeling* 281:1–10.

Van de Weghe, N.; Kuijpers, B.; Bogaert, P.; and Maeyer, P. D. 2005. A Qualitative Trajectory Calculus and the Composition of Its Relations. *GeoSpatial Semantics* 281:60–76.

Vieira, D. C. D. L.; Adeodato, P. J. L.; and Gon, P. M. 2010. Improving Reinforcement Learning Algorithms by the Use of Data Mining Techniques for Feature and Action Selection. *IEEE International Conference on Systems Man and Cybernetics* 1863–1870.

Zimmermann, K., and Freksa, C. 1996. Qualitative spatial reasoning using orientation, distance, and path knowledge. *Applied Intelligence* 6(1):49–58.