Effects of Training Data Variation and Temporal Representation in a QSR-Based Action Prediction System

Jay Young and **Nick Hawes**
Intelligent Robotics Lab, School of Computer Science,
University of Birmingham
UK

Abstract

Understanding of behaviour is a crucial skill for Artificial Intelligence systems expected to interact with external agents – whether other AI systems, or humans, in scenarios involving co-operation, such as domestic robots capable of helping out with household jobs, or disaster relief robots expected to collaborate and lend assistance to others. It is useful for such systems to be able to quickly learn and re-use models and skills in new situations. Our work centres around a behaviour-learning system utilising Qualitative Spatial Relations to lessen the amount of training data required by the system, and to aid generalisation. In this paper, we provide an analysis of the advantages provided to our system by the use of QSRs. We provide a comparison of a variety of machine learning techniques utilising both quantitative and qualitative representations, and show the effects of varying amounts of training data and temporal representations upon the system. The subject of our work is the game of simulated RoboCup Soccer Keepaway. Our results show that employing QSRs provides clear advantages in scenarios where training data is limited, and provides for better generalisation performance in classifiers. In addition, we show that adopting a qualitative representation of time can provide significant performance gains for QSR systems.

Introduction

For robots to work in multi-agent situations, from sports to search and rescue, it is important that they are able to understand and predict the behaviour of the other agents. This is true whether the other agents are humans or other robots. In some highly cooperative scenarios this ability may be provided by direct access to the states and plans of the other agents (e.g. when acting as part of a cooperative team of centrally coordinaed agents). However, in many other situations (such as in ad-hoc teams (Stone, Kaminka, and Kraus 2010), or when working alongside humans) such direct access is not possible. Instead the robot in question must be able to predict the behaviour of the other agents from prior experience and the knowledge it has about them. This problem of behaviour prediction from prior experience is the one addressed in this paper. In particular we focus on spatially-situated behaviour where past spatial events (e.g. movements in space) can serve as a guide to future behaviour. We evaluate our work in the RoboCup simulation domain, but the contributions are applicable to any problem domain which features such spatially-situated behaviour.

In previous work we demonstrated that by encoding the behaviour of agents in *qualitative representations* we were able to out-perform prediction based on purely metric information (Young and Hawes 2013). This is due to the inherent generalisation abilities of qualitative representations, plus the smaller state space delivered by qualitative abstraction. Our long-term aim is to create a framework which allows a robot to learn how to behave (at a high level) in multi-agent tasks based on the observations of other agents. We wish do to this by including the as little domain knowledge as possible, thus creating a system applicable to a wide variety of systems with minimal user input. As such our system uses multiple related qualitative representations to create a purposely large qualitative state representation (in order that domain-specific contents are not lost during abstraction), then applies dimensionality reduction and machine learning techniques to learn predictions from observations of this representation.

This paper continues the development of our framework by making two further contributions:

- A comparison of how a variety of learning mechanisms perform on qualitative and metric representations when using two different, implicit, representations of time;
- A comparison of how the performance of these mechanisms varies given the availability of training data.

Related Work

Qualitative representations (QRs) have been used previously in learning tasks across of a variety of domains. For example, for learning and recognition of events in an aircraft loading domain (Sridhar and Cohn 2010) and desktop activities (Behera, Cohn, and Hogg 2012), and for reinforcement-learning for robot navigation (Frommberger 2008). The problem of behaviour modelling in RoboCup has been tackled in a number of ways including reinforcement learning (Vieira, Adeodato, and Gon 2010; Jung and Polani ; Shimada, Takahashi, and Asada 2010; Molineaux, Aha, and

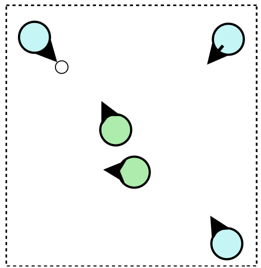

Figure 1: A 3vs2 Keepaway Scenario

Sukthankar 2008), and case-based reasoning (Ros, Llu, and Mantaras ; Floyd, Esfandiari, and Lam 2008). Little work investigates the application of qualitative, relational representations to the problem. Bombini et al. (2010) characterised player behaviour using relational sequence models using pre-defined relational symbols over actions such as dribble, shoot and pass. Molineaux, Aha, and Sukthankar (2008) employ a similar abstraction, just considering co-operative behaviour acquisition.

RoboCup Simulation

The RoboCup simulation domain is a widely-used multi-agent testbed which simulates a game of soccer played by individual playing agents (players) in 2D. At any point in time, all players have a 2D pose (i.e. a 2D position and orientation) and the ball has a 2D position. Given limited sensing of the surrounding players and the ball, each player must choose an *action*: *kick*, *dash* or *turn*. The simulation provides access to the metric positions of all game elements (including static features such as the position of the goals) at every frame, plus the actions the agents took. We refer to this data as the game's *state*. The problem we address in this paper is predicting the future actions of agents given a sequence of preceding states. This is a standard problem in the literature (Vieira, Adeodato, and Gon 2010; Jung and Polani ; Shimada, Takahashi, and Asada 2010; Molineaux, Aha, and Sukthankar 2008; Ros, Llu, and Mantaras ; Floyd, Esfandiari, and Lam 2008).

In this paper we address the *keepaway* sub-game (Stone et al. 2006). Keepaway features two competing teams, one trying to maintain control of the ball, the other trying to gain control of the ball. Figure 1 shows a typical 3v2 keepaway scenario. The keeper team is situated around the borders of the keepaway area, with the opposing team in the centre. The game is split into slices called episodes, wherein a single episode continues so long as the keeper team controls the ball. As soon as control is lost to the other team (or the ball leaves the keepaway area), a new episode starts. In this work we make use of third-party benchmark agents that play the keepaway task[1]. These agents follow hand-coded policies, and provide a baseline level of performance. Our intent is to predict the behaviours of these benchmark agents in the 3v2 game configuration.

[1] http://github.com/tjpalmer/keepaway

Qualitative Representations

In order to provide a more compact representation of the game's (metric) state for learning, we abstract it using *qualitative spatial representations* (QSRs) (Cohn and Hazarika 2001), i.e. mapping specified ranges of quantitative inputs into qualitative symbols (Frommberger and Wolter 2010). As player behaviour in soccer is mostly based on the interactions between multiple game entities, we focus particularly on (binary) *relational* representations, i.e. the relative positions of two game entities (e.g. two players, one player and the ball etc.).

The understand the intuition behind the use of qualitative relational representations in a prediction task such as ours, consider the situation of a player passing to teammate, with an opposition player trying to block the pass. Instances of this situation will happen all over the pitch in many different player configurations. At the quantitative level (2D poses) each instance will appear different to an observer. With an appropriate qualitative abstraction (i.e. capturing the relative positions of the agents) these instances can all be treated as examples of similar, or the same, type of situation. In a learning task, such an abstraction provides the learner with more examples of the situation (compared to the metrically dissimilar examples), aiding knowledge transfer and generalisation, and improving the scalability of learning systems (Frommberger 2010).

As we wish our system to ultimately choose its own appropriate abstraction of the state it is using to predict actions, we apply multiple related QSRs to the metric state to generate a *qualitative state*. The QSRs we use are the *Region Connection Calculus* (RCC) (Randell, Cui, and Cohn 1992), the *Qualitative Trajectory Calculus* (QTC) (Van de Weghe et al. 2005), and the *Star Calculus* (Renz and Mitra 2004). The following paragraphs provide a brief overview of these calculi and our application of them in the RoboCup simulation domain. For a more detailed description please see (Young and Hawes 2013).

RCC is used to represent *connectedness* relations between spatial regions. In our work we apply the RCC5 subset of RCC which 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; and *proper part inverse* (**PPi**) – one region is entirely enclosed by the other.

For a full game, we encode the spatial configuration of a football pitch a set of rectangular regions, representing areas such as the left and right halves, penalty areas etc. In keepaway, we only represent the keepaway area. Players and the ball are represented as their minimum bounding rectangular regions. We exhaustively calculate RRC5 relations between all pairs of regions. Transitions in RCC5 relations between pairs of game entities are governed by a *conceptual neighbourhood* which encodes the set of valid transitions (Gooday and Cohn 1994). We use this neighbourhood when comparing states expressed in terms of QSRs.

QTC describes the relative movement of objects. For this study, we employ the QTC_B (Basic) relation set, which

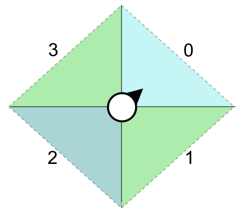

Figure 2: A set of four angular zones egocentric with respect to a player, such that the 0^{th} zone is always aligned with the orientation of the player's body.

encodes the following relations between 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 (-)/towards (+)/is stable with respect to (0) *l*.
Left/Right relation: *k* is moving to the left of (-)/right of (+)/is stable with respect to (0) *l*.
Relative Motion relation: *k* is moving faster than (-)/slower than (+)/is stable with respect to (0) *l*.

Similarly to RCC, transitions between QTC relations are also 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 apply the QTC calculus to all *mobile* entities, such as players and the game ball.

The Star calculus describes the qualitative direction between points in space with respect to one-another using a set of binary relations (Renz and Mitra 2004). Star employs angular zoning based on either an adjustable granularity parameter, or by specifying a sequence of angles. The result is a set of circle sectors emanating from a point, extending into infinity, discretising over a range of angles. The Star relation between two points then is determined by taking the angle between them and determining which zone the result falls in to. In our work, the Star relations that hold between all mobile entities are calculated on each time step. As seen in Figure 2, we ensure that Star relations are fixed with reference to the agent's heading, that is, the set of angular zones is *egocentric* with a given agent.

Temporal Information

In dynamic, spatially-situated domains such as RoboCup, it is important to include changes to space *over time* in the state representation, as agent behaviour can depend on more than just the current state. There are a number of different ways this can be done. The simplest way is to extend the state representation to include a vector of M previous states in addition to the most recent state. However, this has the potential to include sequences of states which do not change qualitatively. A second approach, taken by for example (Sridhar, Cohn, and Hogg 2010), is to compress qualitatively identical states into a single state. This then allows reasoning over M different states, capturing more information in the same size state vector.

Kick	Dash	Turn
11%	22%	66%

Figure 3: Distribution of actions in training set

These two sequence-based approaches encode time implicitly, but it is also possible to include time explicitly in the state. Many qualitative approaches use Allen's interval algebra (Allen 1983), which allows explicit qualitative reasoning about temporal relations between events. QTC also captures temporal information as the value of its relationships are calculated over a time window prior to the state they occur in. In this paper we explore the effect of utilising these different temporal representations – with the exception of Allen's algebra – in a supervised learning approach. We have not used Allen's algebra yet as we do not predict it will add much beyond compressed qualitative sequences in our learning approach. This would change if a different reasoning approach was employed, or something different was being predicted.

Behaviour Prediction Task

Our problem can be formally stated as follows. Given an observed state (which may be composed of a set of QSRs, or be a set of continuous metric values), we wish to be able to predict which action (from the finite set of actions A) a given player (from the set of all players X) will take in the given state. We consider an observed *sequence* of previous states, and so given a vector M of previous state observations, our goal is to predict the action $a \in A$ to be taken by player $x \in X$ at the current simulation step t. We employ different learning approaches and representations to tackle this problem. For all learning approaches, our training sets consist of a state representation and the actions each agent performs in that state. Actions in RoboCup may have continuous parameters, such as the power of a kick or the angle of a turn, meaning that an action is composed of a discrete label and a continuous parameter. For now, we focus on predicting the *label* of actions, ignoring the prediction of parameters, following the same approach as the work of (Floyd 2013).

Experimental Setup

We utilised the benchmark agents of Stone et al. to generate data sets on a 3vs2 configuration of the keepaway task, producing a data set of 1000 total episodes of agents following default policies. We perform 10-fold cross-validation on this data, ensuring that episodes do not become fragmented across fold boundaries. That is, each fold contains a discrete set of complete episodes.

In RoboCup soccer (and similar domains) actions may be unequally distributed – that is, some actions may occur more frequently than others – Figure 3 shows the distribution of actions in our training set across all episodes. This makes the learning task challenging, as a system must be able to predict things such as Kick actions, which occur very rarely in the training set.

We now briefly overview the learning approaches used in our experiments.

k-Nearest Neighbours with Conceptual Neighbourhood Distance

As a distance measure for the kNN algorithm, we utilise our *Conceptual Neighbourhood Distance* (CND) measure. When considering the distance between relations in any QSR state representation, we must obey the constraints of the *conceptual neighbourhoods* which govern transitions between relations (Cohn 1995). We can think of a QSR state as simply a binary string, and as such we may use a naieve approach to comparing strings such as a Hamming distance (Hamming 1950). In the case of qualitative calculi relations are not equidistant, and transitioning between two states may require transitioning through an intermediate state first – a Hamming distance would regardless count such a transition as a distance of 1, rather than 2 if we take into account the underlying semantics of QSRs. We encode the conceptual neighbourhood as transition-cost matrix, which allows us to achieve a smoother, more accurate description of the distance between QSR states. In our results, we show both the performance of a kNN algorithm utilising our CND distance, as well as one utilising a standard Hamming distance. Both with $k = 3$

Bayesian Classifier

We employ a Bayesian Classifier which predicts $P(a|F)$ where $a \in A$ the set of actions, and F is some observed set of features over a finite horizon of previous states. Our prediction is then made by the maximum a posteriori decision rule.

In our work, we employ a feature selection mechanism to determine which relations are most salient to predicting about certain actions. That is, we are interested in the *relative importance* of relations that appear in the state representation, and determining which relations are most strongly linked with particular actions. The behaviour of our agents is spatially-grounded and situated, and as such there will be some spatial relations that influence action selection more than others. But, we would rather not engage in knowledge engineering to encode this (requiring that we know *a priori* which relations to weight, and how – which we may not), and would prefer to discover these relationships from the data, in order for our system to remain domain agnostic. Our approach to this is to make use of a single-link agglomerative hierarchical clustering algorithm (Honkela, Seppä, and Alhoniemi 2008; Sibson 1973), which uses the correlation between relation variables and action classes as a distance measure. We say that clusters can be merged if they statistically correlate with the same class, and we merge those clusters together first that correlate most strongly with a class. This allows us to explore the space of *relation sub-sets*, since it may be the case that a relation (or set of relations) when combined with another relation or set becomes more useful than if it were present on its own. Discovering such relationships may provide us with valuable predictors by uncovering hidden structure in the data. The algorithm then provides us the probability distribution over features used by the Bayesian classifier.

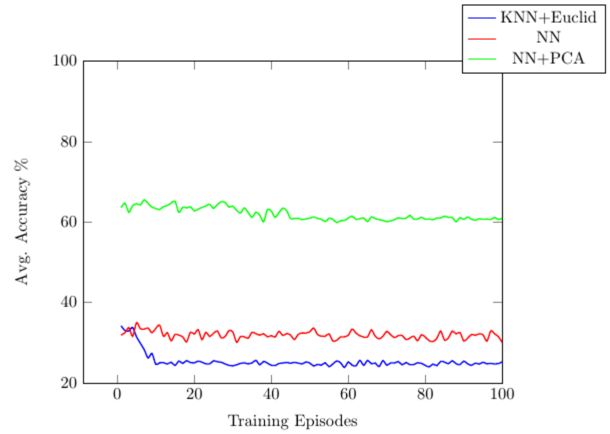

Figure 4: Metric Results with simulation time

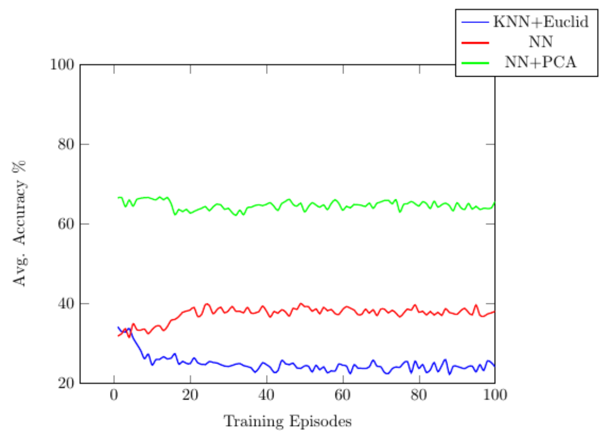

Figure 5: Metric Results with collapsed time

Metric Approaches

We make use of a kNN approach, a standard feed-forward Neural Network and a NN with a pre-processing step of Principle Components Analysis on a metric representation of our problem. For the kNN distance measure we employ a Euclidian distance measure between metric states. We also employ a method of collapsing time in metric representations, whereby states are compressed into a single representation if their Euclidian distance is less than some parameter D, which we specify as the minimum distance that must exist between two states before they can be considered distinct. This we arbitrarily choose to be a difference of 15%.

Temporal Information Results

We report our results as follows. Each point on the graph represents the average accuracy of a system given a data set of a given size, shown on the horizontal axis, and performing

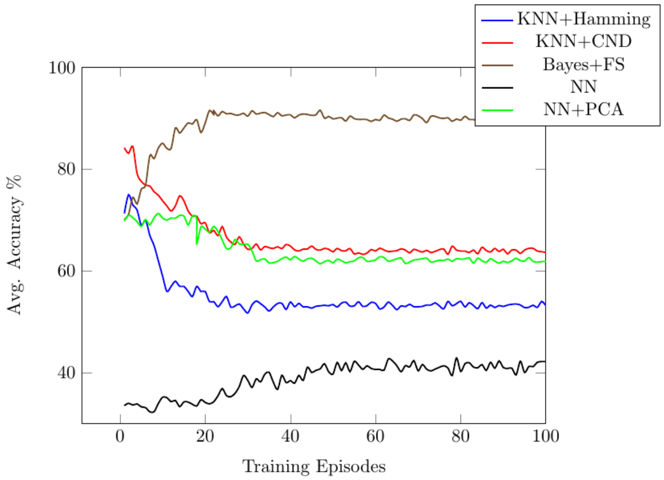

Figure 6: Simulation Time-based predictions using QSRs

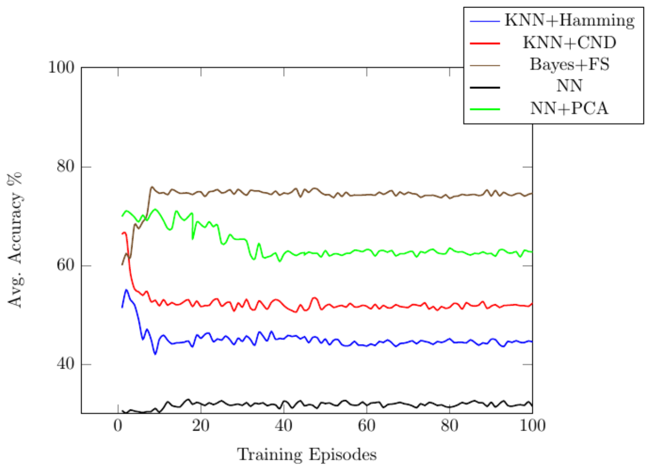

Figure 7: Collapsed-time-based predictions without QTC calculi

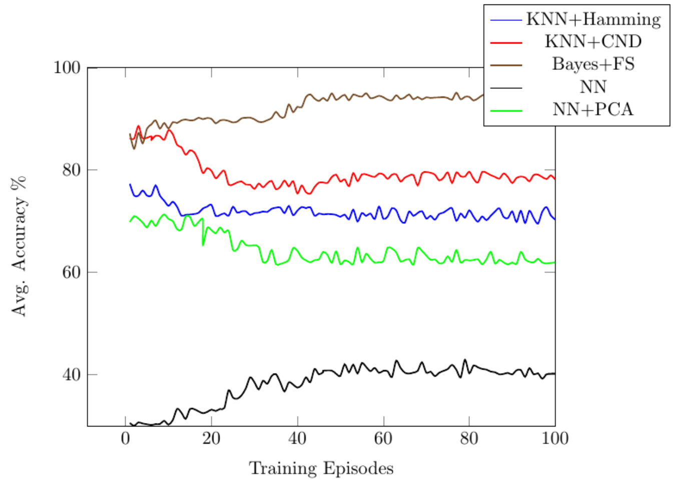

Figure 8: Collapsed-time-based predictions with QTC calculi

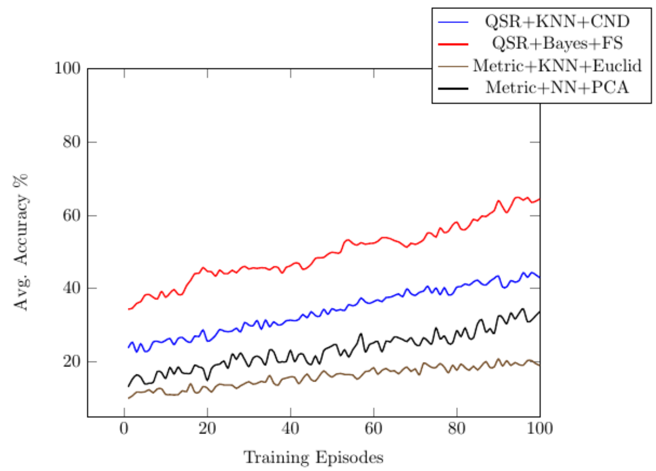

Figure 9: Training Data Generalisation Results

10-fold cross-validation on that dataset alone. We report our results up to 100 training episodes, since this is where the performance graphs roughly converge to a steady state.

Figure 6 shows the results of our experiments using simulation time – that is, the natural, discrete, time steps produced by the simulator.

Figure 7 shows the result of our system utilising the collapsed time approach of Sridhar, Cohn, and Hogg, however excluding the Qualitative Trajectory Calculus from the state representation.

Figure 8 shows the results of utilising the collapsed-time approach with the QTC calculus included in the state representation.

Training Data Results

For our next experiments, we analyse how well our systems are capable of generalising from increasing amounts of training data. For each experiment, we perform a two-fold cross-validation on the training data, however instead of partitioning the data into two equally sized sets, we partition such that the training set contains some n number of training examples, where n is initially 1. We then use those n training examples to construct each classifier, and employ them to make predictions on the contents of the second, unseen fold. We perform this with both the QSR and metric-based systems on a selection of our classifiers.

Each point is therefore the average performance of the system after constructing the classifier with k training instances, and validating on $1000 - n$ validation instances. We show results up to 100 training instances for brevity.

Figure 10 shows the overall accuracy of each system at $n = 500$, which corresponds to standard 2-fold cross-validation, whereby half the training set is used to construct the classifier, and the other half is used as a validation set. This shows a slice through the graph in Figure 9.

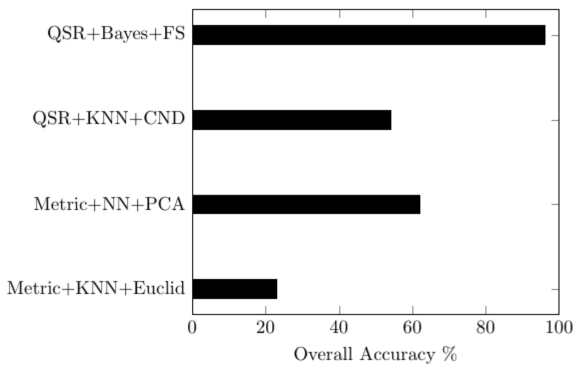

Figure 10: Training Data Generalisation Results at $n = 500$ (2-fold cross-validation)

Discussion

Our results show that learning systems utilising QSR-based representations outperform those employing quantitative, metric representations. This is especially apparent when the systems utilising QSRs are provided a limited amount of training data. QSRs compress a potentially large range of quantitative observations into a single symbol – even if they have not been observed directly. Entirely metric-based systems must be explicitly provided this information in the training set, or must engage in some pre-processing step to generalise what has been presented. We observe that for small amounts of training data, simple, nearest-neighbours approaches perform with a reasonable degree of accuracy in the collapsed-time scenario. However, as the size of training sets increases, these classifiers become exposed to more variation, meaning that the classifiers will have a larger number of qualitatively similar states to pick from. This may not be particularly beneficial – a state that is similar qualitatively may be, considering the underlying semantics of the domain, distant. Or, states may be close because they have only small qualitative differences, but these may be irrelevant to predicting about the actions of a particular agent – i.e. a nearest neighbour is found, but for practical purposes is the same as the original. As mentioned previously, we know that for each agent a given set of spatial relations will be more important than others. This is the reason why our approach utilising a Bayesian classifier and a feature selection mechanism improves as more training data is provided, as it is capable of determining the relative importance of relations and sets of relations to each class. As such, the more data it is provided, the more it is able to determine which are the most salient relations, and which function as background noise. In short, the difference between the two approaches can be thought of as learning to look for similarity in the right *dimensions* rather than in the overall state vector, as in the nearest neighbours approaches.

Particularly interesting is the effect of collapsing time, ensuring the difference between any two adjacent time steps is always qualitatively different, rather than allowing for repetitions of the same state, as in simulation time. This allows us to capture sequences in the data that might occur over varying time horizons. We may not know what these horizons might be, so detecting the sequences may be challenging. By collapsing time we restrict our attention to features observed the n most recent qualitatively different states when predicting $P(a|F)$, and so regardless of whether a sequence occurs over a varying time horizon, collapsing time allows for easier detection by its characteristic qualitative change.

In the RoboCup domain we also observe that the presence of the QTC calculi in our state representation provides a significant boost in prediction accuracy, shown in Figures 8 and 7, highlighting the need to represent motion information when dealing with highly dynamic domains. The highest performing configuration of our system is the one that both collapses time and utilises the QTC calculus, which has also been employed successfully by (Hanheide, Peters, and Bellotto 2012) in robot navigation tasks.

Conclusion

We presented our work on analysing the advantages that may be provided by QSRs to machine learning systems in terms of reducing the amount of training data required, as well as an analysis of different representations of time that might be possible. We showed that utilising a QSR-based representation affords systems the ability to generalise from a smaller number of training examples compared to those utilising metric representations. We also showed the effects of varying the temporal representation in systems by collapsing time from simulation time to time based on qualitative change. Not only does this improve performance, but also reduces the volume of training data required by a system, since adjacent identical states are collapsed into a single state. We argue that this allows for easier discovery of temporal sequences that occur along varying time horizons.

Our particular approach however poses new problems for learning systems. In our work, we compared a nearest-neighbours based approach to ones utilising feature-selection mechanisms, and showed that performance can be improved by considering the relative importance of individual QSRs, as well as sets of QSRs, with the targets of prediction. Our future work now looks at how these advantages can be used by a system intended to *act* within the domain alongside team-mates. In Figure 10 we see that the system utilising QSRs+KNN+CND converges to an accuracy of around 58%. The system utilising Metric+NN+PCA however takes a significant amount of time to train, due to the pre-processing step of PCA. Our approach with the Bayesian classifier and feature selection similarly has a cost in terms of pre-processing time. That is, in these cases it is possible to improve predictive accuracy by applying a pre-processing step. However, one question we consider to be open, and which we intend to investigate next, is the relative benefit of such improvements in terms of practical performance. It may be the case that we can achieve a reasonable, baseline level of performance from an agent acting based on learned models and a kNN approach with no pre-processing. The relationship between predictive accuracy and practical performance is one that remains to be fully explored.

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.

Behera, A.; Cohn, A.; and Hogg, D. 2012. Workflow activity monitoring using dynamics of pair-wise qualitative spatial relations. *Advances in Multimedia Modeling*.

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.

Cohn, A., and Hazarika, S. 2001. Qualitative spatial representation and reasoning: An overview. *Fundamenta Informaticae*.

Cohn, A. 1995. The challenge of qualitative spatial reasoning. *ACM Computing Surveys (CSUR)* 44(113):1–5.

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.

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*.

Floyd, M. 2013. A General-Purpose Framework for Learning by Observation. (May).

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

Frommberger, L. 2008. Learning To Behave in Space: a Qualitative Spatial Representation for Robot Navigation With Reinforcement Learning. *International Journal on Artificial Intelligence Tools* 17(03):465–482.

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*.

Hamming, R. 1950. Error detecting and error correcting codes. *Bell System technical journal*.

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.

Honkela, A.; Seppä, J.; and Alhoniemi, E. 2008. Agglomerative independent variable group analysis. *Neurocomputing*.

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.

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*.

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.

Sibson, R. 1973. SLINK: an optimally efficient algorithm for the single-link cluster method. *The Computer Journal*.

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

Sridhar, M.; Cohn, A.; and Hogg, D. 2010. Relational Graph Mining for Learning Events from Video. *AAAI* 1–6.

Stone, P.; Kuhlmann, G.; Taylor, M.; and Liu, Y. 2006. Keepaway soccer: From machine learning testbed to benchmark. *RoboCup 2005: Robot Soccer* ....

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.

Young, J., and Hawes, N. 2013. Predicting Situated Behaviour Using Sequences Of Abstract Spatial Relations. *In Proceedings of the AAAI 2013 Fall Symposium How Should Intelligence be Abstracted in AI Research: MDPs, Symbolic Representations, Artificial Neural Networks, or _____?*