

# Autonomous learning for a cognitive agent using continuous models and inductive logic programming from audio-visual input

D. Magee, C.J. Needham, P. Santos, A.G. Cohn and D.C. Hogg

School of Computing,  
University of Leeds,  
Leeds,

LS2 9JT, UK

{drm/chrisn/psantos/agc/dch}@comp.leeds.ac.uk

## Abstract

A framework for autonomous (human-like) learning of object, event and protocol models from audio-visual data, for use by an artificial “cognitive agent”, is presented. This is motivated by the aim of creating a synthetic agent that can observe a scene containing unknown objects and agents, operating under unknown spatio-temporal motion protocols, and learn models of these objects and protocols sufficient to act in accordance with the implicit protocols presented to it. The framework supports low-level (continuous) statistical learning methods, for object learning, and higher-level (symbolic) learning for sequences of events representing implicit temporal protocols (analogous to grammar learning). Symbolic learning is performed using the “Progol” Inductive Logic Programming (ILP) system to generalise a symbolic data set, formed using the lower level (continuous) methods. The subsumption learning approach employed by the ILP system allows for generalisations of concepts such as equality, transitivity and symmetry, not easily generalised using standard statistical techniques, and for the automatic selection of relevant configurational and temporal information. The system is potentially applicable to a wide range of domains, and is demonstrated in multiple simple game playing scenarios, in which the agent first observes a human playing a game (including vocal facial expression), and then attempts game playing based on the low level (continuous) and high level (symbolic) generalisations it has formulated.

## Introduction

The perceived world may be thought of as existing on two levels; the sensory level (in which meaning must be extracted from patterns in continuous observations), and the conceptual level (in which the relationships between various discrete concepts are represented and evaluated). We suggest that making the link between these two levels is key to the development of artificial cognitive systems that can exhibit human-like qualities of perception, learning and interaction. This is essentially the classic AI problem of “Symbol Grounding” (Harnad 1990). The ultimate aim of our

work is truly autonomous learning of both continuous models, representing object properties, and symbolic (grammar like) models of temporal events, defining the implicit temporal protocols present in many structured visual scenes. Much work has been carried out in the separate areas of pattern recognition and model building in continuous data (see for example (Duda, Hart, & Stork 2000)) and symbolic learning in various domains such as robotics/navigation (Bryant *et al.* 1999), bioinformatics (Sternberg *et al.* 1994) and language (Kazakov & Dobnik 2003). Several systems have been presented that link low-level video analysis systems with high-level (symbolic) event analysis in an end-to-end system, such as the work of Siskind (Siskind 2000) that uses a hand-crafted symbolic model of ‘Pickup’ and ‘Putdown’ events. This is extended in (Fern, Givan, & Siskind 2002) to include a supervised symbolic event learning module, in which examples of particular event types are presented to the learner. Moore and Essa (Moore & Essa 2002) present a system for recognising temporal events from video of the card game ‘blackjack’. Multiple low level continuous temporal models (Hidden Markov Models), and object models (templates) are learned using a supervised procedure, and activity is recognised using a hand defined Stochastic Context-Free Grammar. A similar approach is used by Ivanov and Bobick (Ivanov & Bobick 2000) in gesture recognition and surveillance scenarios. However, none of these systems is capable of autonomous (unsupervised) learning of both continuous patterns and symbolic concepts. The motivation behind our research is to learn both low level continuous object models and high-level symbolic (grammar like) models from data in an arbitrary scenario with no human interaction. Systems capable of unsupervised learning of both continuous models of image patches and grammar-like (spatial) relations between image patches have been presented by the static image analysis community (e.g. (Aksoy *et al.* 2003)). These involve the use of general (non-scene specific) background knowledge of the type of relations that may be important (e.g near, far, leftof etc.). It is our aim to develop conceptually similar approaches for the analysis of dynamic video data. These would be similar to the grammars used in (Moore & Essa 2002; Ivanov & Bobick 2000), which are currently manually defined.

We separate learning into two parts: i) Low level learning

of patterns in continuous input streams, and ii) High level (symbolic) learning of spatial and temporal concept relationships. This separation of low level and high level processing is motivated by our understanding of the human brain. The visual cortex, and associated parts of the brain, are known to provide initial pre-processing of continuous visual input (Petkov 1995). The structure of this pre-processing develops as a child grows based on experience. This is very much analogous to unsupervised / self-organising pattern recognition methods. There is also evidence, from the (possible) existence of mirror neurons (Stamenov & Gallese 2002), that perception of simple temporal events (e.g. smiling) and the corresponding action are associated at a very low (sub-conscious) level of brain processing, far removed from high level conceptual reasoning and control. In our framework this relationship between simple (vocal) action perception and repetition is explicit. How much this relationship is innate or learned in humans is unclear. However, in our framework the important thing is that this relationship exists at the low level. Currently this relationship is hard coded, however it is possible to learn these type of relationships. In the work of (Fitzpatrick *et al.* 2003), the link between visual perception of action and generation of the same action is learned for a humanoid robot performing simple tasks. Initially the action is performed by a human. The robot subsequently learns to mimic this action by experimentation. It can then copy an action observed at a later time. Such learning would be a valuable addition to our system, but is beyond the scope of this paper.

We propose a learning framework that consists of three elements; an attention mechanism, unsupervised low-level (continuous) object learning, and high-level (symbolic) learning of temporal protocols. Egocentric learning is carried out, meaning the models built are based on linking the behaviour of an agent (e.g. a vocal utterance) to the observed scenario, rather than being holistic models of the complete scenario. This allows the models to easily drive the behaviour of a synthetic agent that can interact with the real world in a near-natural way. Multiple derived features for each object identified by the attention mechanism are grouped into semantic groups representing real-world categories such as position, texture and colour. Clusters are formed for each semantic feature group separately using a clustering algorithm. Classifying models are then built using cluster membership as supervision. These models allow novel objects (identified by the attention mechanism) to be assigned a class label for each semantic group (texture, position, etc.). These symbolic labels are augmented by an annotation of the corresponding vocal utterances of the player(s), and used as input for symbolic learning (generalisation) based on the Progol Inductive Logic Programming system (Muggleton 1995). The output of the continuous classification methods can be presented in such a way that instances of concepts such as equality, transitivity, symmetry etc. may be generalised, in addition to generalisations about the protocols of temporal change. The advantage of Progol’s learning approach is that learning can be performed based on noisy (partially erroneous) data, using positive examples only.

Our prototype implementation has been applied to the learning of the objects, and protocols involved in various simple games including a version of “Snap”, played with dice, and a version of the game “Paper, Scissors, Stone” played with cards. Typical visual input is shown in Figure 1. It is a common argument (Hargreaves-Heap & Varoufakis 1995) that many real-world social interaction scenarios may be modelled as games, which suggests our system is applicable beyond this domain. We would make this argument.

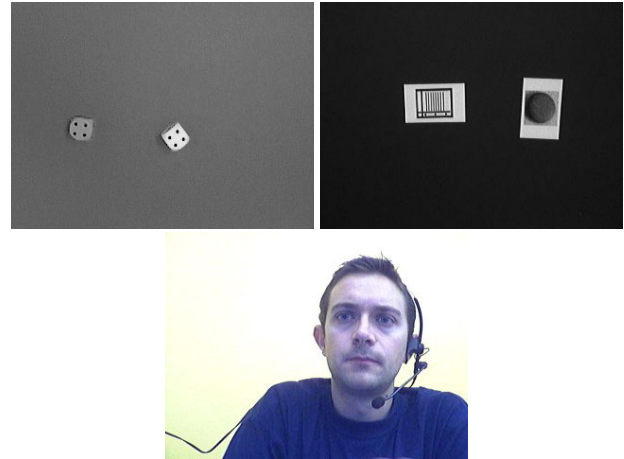


Figure 1: Typical input data

## Learning framework

Our framework divides learning into two parts; low-level (continuous) object learning, and high-level (symbolic) protocol and relationship learning. To facilitate autonomous (fully unsupervised) learning, a spatio-temporal attention mechanism is required to determine ‘where’ and ‘when’ significant object occurrences and interactions take place within the input video stream of the scenario to be learned from. The object models produced during low-level learning are used to produce a symbolic stream for use in the high-level learning. This symbolic stream is augmented with the vocal utterances issued by the player(s) participating in the game. These vocal utterances may either take the form of passive reactions (e.g. “snap”), or active statements of intent (e.g. “pickup-lowest”). The latter generates an implicit link between the vocal utterance and the subsequent action in the data stream. Our high-level system can learn this link, and thus an agent based on the learned model can generate these utterances as a command to actively participate in its environment (as currently our framework is implemented on a software only platform, with no robotic component). It should be noted that this approach relies on a direct link between the perception of a given vocal utterance and the generation of this utterance by the agent. In the current implementation of our framework the vocal utterance is “perceived” by the agent via hand annotation of facial video sequences, and thus the link between the agents perception and generation of an action is trivial. Automation of this

process could be performed using standard speech recognition software, with the link between action perception and action generation (generation of speech using a standard speech generator) being made via a pre-defined vocabulary of words. Our eventual aim is to learn our own vocabulary of utterances autonomously from the audio-visual face data. Such a system would have to make its own link between action perception and action generation. This is the subject of current research, but is beyond the scope of this paper. Figure 2 provides an overview of our learning framework.

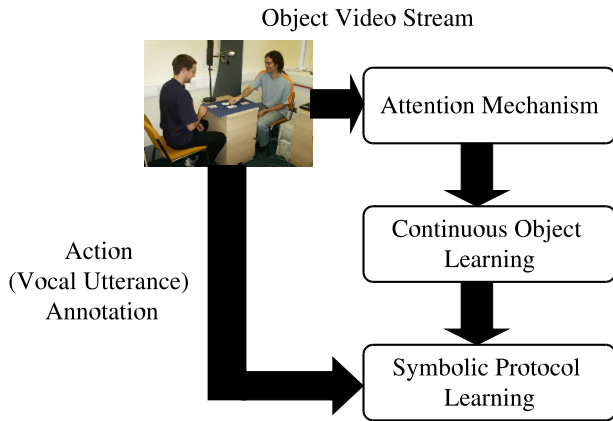


Figure 2: Overview of the learning framework

It should be noted that conceptually the framework does not limit the perception and generation of action to vocal utterances; however a link is required between the perception and generation of individual agent actions for learned models to be used in an interactive agent. Vocal utterances are a good example of an action that can be perceived and generated without specialised hardware. It was for this reason they were chosen for our example implementation. The remainder of this section will be divided into sub-sections on attention, continuous object model learning, and higher level symbolic learning using Inductive Logic Programming.

### Spatio-temporal attention

Video streams of dynamic scenes contain large quantities of data, much of which is irrelevant to scene learning and interpretation. An attention mechanism is required to identify ‘interesting’ parts of the stream, in terms of spatial location (‘where’) and temporal location (‘when’). For autonomous learning, models or heuristics are required to determine what is of interest, and what is not. Such models could be based on motion, novelty, high (or low) degree of spatial variation, or a number of other factors. In our framework it is merely important that an attention mechanism exists to identify interesting areas of space and time. For this reason we have chosen to use motion in our example implementation, as this is straightforward to work with. We make no claim that attention based on motion only is suitable in all scenarios, however it is appropriate in our chosen domains. It is highly likely that no single factor could provide a generic at-

tention mechanism for learning and interpretation in all scenarios. In the view of the authors it is much more likely that multiple attention mechanisms would be required for fully generic learning.

The spatial aspect of our attention mechanism is based around a generic blob tracker (Magee 2004) that works on the principle of multi-modal (Gaussian mixture) background modelling, and foreground pixel grouping. This identifies the centroid location, bounding box and pixel segmentation of any separable moving objects in the scene in each frame of the video sequence. If multiple objects are non-separable from the point of view of the camera they are tracked as a single object, until such time as they are separable. This is not a significant drawback in the example scenarios we present in this paper (and many others), however there are situations where a more complex spatial attention method would be required.

The temporal aspect of our attention mechanism identifies key-frames where there is qualitatively zero motion for a number of frames (typically 3), which are preceded by a number of frames (typically 3) containing significant motion. Motion is defined as a change in any objects’ centroid or bounding box above a threshold value (typically 5 pixels, determined from observed tracker positional noise). This method for temporal attention is based on the assumption that all objects remain motionless following a change in state (and that the process of change is not in itself important). This is valid for the example scenarios we present within this paper, however we are actively researching more complex temporal attention mechanisms that do not make these assumptions.

### Continuous object learning and classification

In autonomous learning it is not in general possible to know *a-priori* what types of visual (and other) object properties are important in determining object context within a dynamic scene. For this reason the use of multiple (in fact large numbers of) features such as colour, texture, shape, position etc. is proposed. We group sets of features together into manually defined semantic groups representing texture, position etc.<sup>1</sup> In this way (initial) feature selection within these semantic groups is performed during continuous learning, and feature selection and context identification between the groups is performed during the symbolic learning stage.

For each semantic group a set of example feature vectors is partitioned into classes using a graph partitioning method (an extension of (Strehl & Ghosh 2002)), which also acts as a feature selection method within the semantic group (see (Magee, Hogg, & Cohn 2003) for details). The number of clusters is chosen automatically based on a cluster compactness heuristic. In other work (Santos, Magee, & Cohn 2004) the number of clusters is deliberately selected as overly large and cluster equivalence is determined during symbolic learning. This will be our preferred approach

<sup>1</sup>In this paper we use a 96D rotationally invariant texture description vector (based on the statistics of banks of Gabor wavelets and other related convolution based operations), and a 2D position vector only.

in future work, as temporal context (in addition to spatial appearance information) is taken into account.

Once a set of examples is partitioned, the partitions may be used as supervision for a conventional supervised statistical learning algorithm such as a Multi-layer perceptron, Radial Basis Function or Vector Quantisation based nearest neighbour classifier (we use the latter in our implementation). This allows for the construction of models that encapsulate the information from the clustering in such a way that they can be easily and efficiently applied to novel data. These models are used to generate training data suitable for symbolic learning. For each object identified by the attention mechanism, a property is associated with it for each semantic group. For example:

```
state([obj0,obj1],t1).
property(obj0,tex0).
property(obj1,tex1).
property(obj0,pos1).
property(obj1,pos0).
```

indicates that there are two objects present at time  $t_1$ . The first belongs to texture class  $tex_0$  and position class  $pos_1$ , and the second to texture class  $tex_1$  and position class  $pos_0$ . These symbolic streams are a good representation of the input stream, however they are not noise free. A fuller explanation of the symbolic representation used is given in section .

### Symbolic learning using Inductive Logic Programming

The previous sections described how models are learned that can convert continuous sensory input into a symbolic data stream in an unsupervised way. We also wish to learn models of the spatio-temporal structure of the resultant (possibly noisy) symbolic streams obtained. I.e. we wish to learn a model of any implicit temporal protocols presented by the scene. (This is directly analogous to learning the grammar of a language by example.) Structure in such streams differs greatly from the structure learned by our lower level processes, in that the data consists of variable numbers of objects (and thus a variable length list of state descriptions is available). In addition, concepts such as equality, symmetry and transitivity exist in such streams. Purely statistical learning methods, such as those used for lower level learning, are not well suited to learning such concepts. We employ a subsumption data generalisation approach, implemented as the Progol Inductive Logic Programming system (Muggleton 1995). Progol allows a set of noisy positive examples to be generalised by inductively subsuming the data representations by more general data representations/rules (with the aim of reducing representational complexity, without over-generalising). Crucial in any inductive learning approach is the way in which data is represented. Progol aims to reduce representational complexity using a search procedure. In realistic scenarios a search of all possible data representations is not possible, and Progol must be guided by rules that define the general form of the solution, and a

suitable presentation of the data to be generalised. We represent the data in a scenario independent/neutral form using the generally applicable symbolic concepts of i) time points ( $time()$ ), ii) object instances ( $object()$ ), iii) object properties ( $proptype()$ ), iv) actions/events ( $actiontype(),actionparameteretype()$ ), and v) relations between i-iv). Each object instance is unique in space and time<sup>2</sup>. Relations used in this work are: temporal succession ( $successor(t_2,t_1)$ , indicating  $t_2$  directly follows  $t_1$ ), object-time relations ( $state([obj_0,obj_1],t_1)$ , indicating  $obj_0$  and  $obj_1$  occur at time  $t_1$ ), action-time relations ( $action(act_1,[param_1],t_1)$ , indicating action  $act_1$ , with parameter  $param_1$  occurred at time  $t_1$ ), and object-property relations ( $property(obj_0,p_0)$ , indicating  $obj_0$  has property  $p_0$ ). It is also possible to use object-object relations (e.g.  $rel(leftof, obj_1, obj_2)$ , indicating object  $obj_1$  is to the left of object  $obj_2$ ), however these are not used in this paper. A short example of this representation is given below:

```
proptype(tex2).
proptype(tex3).
proptype(pos1).
proptype(pos2).
actiontype(utterance).
actionparameteretype(roll).
actionparameteretype(pickuplowest).
```

```
time(t10).
object(obj0).
state([obj0],t10).
property(obj0,tex3).
property(obj0,pos1).
action(utterance,[roll],t10).
```

```
time(t20).
successor(t20,t10).
object(obj1).
object(obj2).
state([obj2,obj3],t20).
property(obj1,tex3).
property(obj1,pos1).
property(obj2,tex2).
property(obj2,pos2).
action(utterance,[pickuplowest],t20).
```

The final element required for Progol to generalise the data is a set of instructions (known as ‘mode declarations’) on the general form of the data generalisation required. These mode declarations separately constrain the sorts of predicates in the head and body of generalisations made (and how these predicates may be combined). As we wish to use the generalisation to generate facial be-

<sup>2</sup>In this paper the fact that two object instances at different times are the same object is not explicitly represented. This is not necessary in many learning scenarios. However, it is possible to encode this information using an object-object relation.

behaviour it is desirable to force the generalisation to contain `action(utterance,...)` in the head of all rules, such that generalisations will be of the form:

```
action(utterance,...) :- ....
```

In this way the resultant generalisation can be fed (with very minor, automatable, modification) into a Prolog interpreter as part of a program for an interactive cognitive agent (see later). We currently put little restriction on the form of the bodies of the rules.

The remainder of this section describes various experiments carried out using variations on the approach described. Appropriate vocal utterances are learned by observation of examples of the games.

**Experiment 1** We define a simple, single player, two dice game based on the card game snap. The two dice are rolled one at a time. If the two dice show the same face the player shouts “snap” and utters the instruction “pickup-both”. Both dice are picked up. Otherwise the player utters “pickup-lowest”, and the dice showing the lowest value face is picked up. Before rolling the player utters the instruction “roll-both” or “roll-one”, depending on if there is a dice already on the table. This is illustrated in Figure 3.

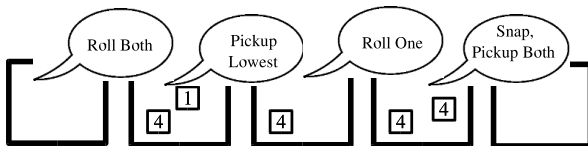


Figure 3: Example of the game used in Exp. 1

**Experiment 2** In this experiment the utterances relating to the game in experiment 1 are made more specific by stating the face of significance as a second utterance (e.g. “pickup three” or “roll six”). Vocal utterances are represented as a one or two parameter utterance (depending on the number of words in the utterance), e.g.  
`action(utterance, [pickup, one], tN) .`  
`action(utterance, [snap], tN) .`  
 An example of this game is illustrated in Figure 4.

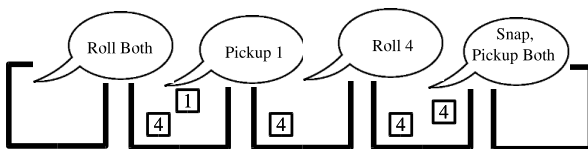


Figure 4: Example of the game used in Exp. 2

**Experiment 3** An alternative game is used based on the game ‘Paper, Scissors, Stone’, in which two players simultaneously select one of these objects. Paper beats (wraps) stone, scissors beats (cuts) paper, and stone beats

(blunts) scissors. Our version of this game is played with picture cards, rather than hand gestures for simplicity. Utterances (‘I win’, ‘draw’ and ‘go’) are represented as a different action for each player. Learning is performed for one player only, and fixed absolute playing positions provide the link between players and cards. E.g. output rules are of the form:

```
action(player1_utterance, [...], tN) :-
  .....
```

Figure 5 illustrates an example of this game.

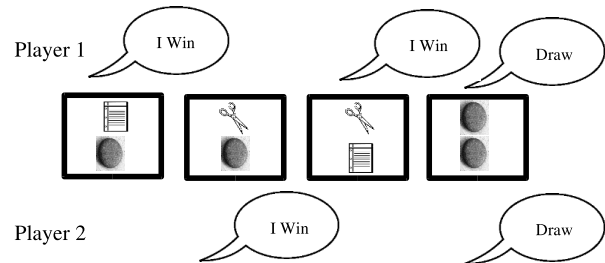


Figure 5: Example of the game used in Exp. 3

### Agent behaviour generation

The rules generated by the ILP learning, and the object models, are used to drive an interactive cognitive agent that can participate in its environment. With a small amount of additional Prolog code this program has been made to take its input from the lower level systems using network sockets, and output its results (via a socket) to a face utterance synthesis module (which simply replays a processed video of the appropriate response). Figure 6 illustrates the operation of the interactive cognitive agent with the objects in the real world scene. A human participant is required to follow the instructions uttered by the synthetic agent (as there is currently no robotic element to our system).



Figure 6: Using learned continuous models and symbolic rules to drive a cognitive agent

Currently the rules produced by Progol (ordered from



most specific to most general if necessary<sup>3</sup>) directly form part of a prolog program. We impose a limit of a single action generation per timestep in the (automatic) formulation of this program. We are working on an rule interpreter which can handle a wider range of scenarios (multiple simultaneous actions, non-deterministic/stochastic outcomes etc.), however this is not necessary for the scenarios presented in this paper.

### Evaluation and results

Several minutes of footage of each game described previously (experiments 1-3) was recorded for training purposes, with separate sequences recorded for evaluation purposes. Training sequences were hand annotated with the actual vocal utterances made. Table 1 gives training and test set sizes.

	# utterances (Training Set)	# utterances (Test Set)
exp 1a/1b	61	35
exp 2a/2b	223	41
exp 3a/3b	176	105

Table 1: Training and test set sizes

Continuous object, and symbolic protocol models were learned from each training sequence and used to drive an artificial cognitive agent. The performance of the agent was evaluated using the (unseen) evaluation sequences. Each experiment was repeated twice, once with a perfect annotation of the vocal utterances for the training sequence (experiment Na), and once with 10% of the utterances randomly replaced with erroneous utterances to simulate error in a speech recognition system (experiment Nb). The number of correct and incorrect utterances generated for the evaluation sequences was recorded for each experiment/model with respect to the actual utterance made (table 2, column 5), and with respect to the utterance that would be expected based on the (possibly erroneous) low-level classification of objects (table 2, column 6). These results are presented in Table 2, with the (intermediate) low-level classification performance (column 4) included for reference.

Although the low-level object classification models are imperfect, a perfect rule-set is generated for experiment 1 when object noise, and when object noise plus utterance noise, is present in the training data. A perfect rule-set is generated for experiment 3 with object noise, however some rules are lost with the introduction of utterance noise. Experiment 2 is more complex, due to the increased utterance possibilities, and so requires more rules than the other two. Some rules are missing in both parts of this experiment, although performance is still reasonable. However, an accurate rule-set for experiment 2 was obtained using noise-free (synthetic) training data, indicating that it is noise in the

<sup>3</sup>In the case that the body of one rule is a specialisation of another, the most general rule is moved below the most specific one in the ordering (if not the case already). This may be determined automatically using a subsumption check on each pair of rule bodies. Otherwise rule ordering is as output by Progol.

Exp.	frames classified completely correctly	correct utterances compared to actual	correct utterances compared to classification
1a	29 (83%)	32 (91%)	35 (100%)
1b	29 (83%)	32 (91%)	35 (100%)
2a	38 (93%)	31 (76%)	32 (78%)
2b	38 (93%)	31 (76%)	32 (78%)
3a	105 (100%)	105 (100%)	105 (100%)
3b	105(100%)	71 (68%)	71 (68%)

Note: Experiment Na: Object identity noise, Experiment Nb: Object identity + Vocal Utterance noise

Table 2: Evaluation results

symbolic data that results in the loss of rules (rather than the structure of the problem). These results demonstrate the graceful degradation of the ILP generalisation with noise. Less general rules are lost, rather than the entire process failing, when noise is introduced. This is essential for future work involving incremental and iterative learning. It is worth examining the rule-set generated by experiment 1 to illustrate the generalisation of the training data performed (Figure 7).

```

action(utterance,[rollboth],A) :- state([],A).
action(utterance,[rollone],A) :- state([B],A).
action(utterance,[pickuplowest],A) :- state([B,C],A).
action(utterance,[snap],A) :- state([B,C],A),
property(B,D), property(C,D).

```

Figure 7: Progol output for experiment 1a

It can be seen from the snap rule in Figure 7 that the concept of property equality has been used in the generalisation of the training data. The rule-set perfectly and concisely represents the protocol of this game, despite errors in the classification of objects in the training data. This may be partially due to most of the erroneous classifications fitting the generalisation, due to the nature of the utterances. It should be noted that the ‘snap’ rule is a specialisation of the ‘pickuplowest’ rule. Currently rules are ordered from most specific to most general for interpretation by the cognitive agent, allowing only the most specific rule to be activated. This is fine for the scenarios presented in this paper; however work has commenced on a stochastic rule-interpreter that selects overlapping rules based on statistics from the training data. This will enable the modelling of more complex situations and non-deterministic outcomes. Figure 8 gives the generalisation from experiment 1b.

It is interesting to note that the generalisation given in Figure 8 is identical to the generalisation in Figure 7, apart from the addition of terms relating to (some of) the erroneous inputs<sup>4</sup>. These extra terms have no effect on the operation of a cognitive agent because, as grounded-assertions, they refer to specific times (which by definition will never recur).

<sup>4</sup>Progol retains these terms so that the generalisation represents the entire data set

```

action(utterance, [rollboth], t600).
action(utterance, [rollone], t663).
action(utterance, [rollboth], t686).
action(utterance, [pickulowest], t902).
action(utterance, [pickulowest], t1072).
action(utterance, [rollboth], t1089).
action(utterance, [rollboth], A) :- state([], A).
action(utterance, [rollone], A) :- state([B], A).
action(utterance, [pickulowest], A) :- state([B, C], A).
action(utterance, [snap], A) :- state([B, C], A),
property(B, D), property(C, D).

```

Figure 8: Progol output for experiment 1b

## Discussion, current and future work

A framework for the autonomous learning of both low level (continuous) and high level (symbolic) models of objects and activity has been presented. It has been demonstrated that a set of object and temporal protocol models can be learned autonomously, that may be used to drive a cognitive agent that can interact in a natural (human-like) way with the real world. The application of this two stage approach to learning means the symbolic representation used is explicitly grounded to the (visual) sensor data. Although our synthetic agent has no robotic capability, it can issue vocal instructions and participate in simple games. The combination of low-level statistical object models with higher level symbolic models has been shown to be a very powerful paradigm. It allows the learning of qualitative concepts and relations such as equality, symmetry and transitivity as well as relative spatial and temporal concepts.

While what is presented in this paper represents a substantial body of work, we are still a long way from where we want to be in terms of developing an agent with true human-like learning and interaction capabilities. Our system currently views the world it perceives as a whole, and cannot compartmentalise different experiences into different categories. As an example, if the training data contained two (or more) different games the system would try to generalise them as a single theory. While this will eliminate a lot of potential redundancy, this may not be the best, or most efficient, way of representing this information. We would like to investigate learning in multiple scenarios, while allowing some generalisation between different scenarios (i.e. an idea of shared concepts between scenarios). We wish to use the non-generalised training instances from Progol output to feedback to, and improve, the lower level object models. In many scenarios this is essential as some objects may not be easily discriminated using spatial appearance alone. In such cases temporal context is essential. The current system is based around single-shot ‘observe and generalise’ learning. In order for temporal information to be usefully included, learning must be extended to be iterative or incremental in nature. This is also an important goal if learning is to be more human-like (human learning continues throughout our entire life). We would like to make this natural extension to our system in due course. An advantage of incremental

learning is that there is an existing model during (much of) the learning phase. This allows learning by experimentation, or “closed-loop” learning. This would require the formulation of a learning goal or motivation (e.g. the desire to map an environment in robotics (Bryant *et al.* 1999)). Our current system has no such explicit motivation. However, the implicit motivation of accurate mimicry could be made explicit. This is an interesting avenue for research.

The practicalities of the ILP approach mean that the presentation of the symbolic data, and the output specification rules, determine the types of generalisations made. Informal experiments have shown us that different rules and input formulations may be required to learn different types of generalisations. How different output rule sets are combined in the context of a cognitive agent is a subject of current research (Santos, Magee, & Cohn 2004). We believe such combination of multiple generalisations is essential if learning in unconstrained scenarios is to be possible. In addition, we are currently building a rule-interpreter that deals with non-deterministic/stochastic scenarios (where a given input results in one of a range of actions) and overlapping rule sets (where one rule takes precedence over another, as in experiment 1). This is based on recording statistics from the training set.

We plan to extend our system to include more object feature types (colour, spatial relationships, global and local shape etc.). It should be possible for the ILP system to learn which object features are relevant in a given scenario.

## Conclusion

We have developed a framework that combines low level (continuous) learning of object and gesture models with high level (symbolic) learning of temporal protocols and conceptual relationships using Inductive Logic Programming. We believe this is the first application of this form of symbolic learning to visual protocol learning. Our prototype system has been applied to learning of the multiple elements of various simple games. The models learned are used to drive a synthetic cognitive agent that can interact naturally with the world. Currently, once learning is performed, the cognitive agent acts in a fully autonomous way. We aim to make our system fully autonomous in the near future, and are developing an audio-visual facial gesture learning system to remove the requirement for manual vocal utterance annotation. We have shown the combination of continuous and symbolic models to be a powerful paradigm which will open up many further avenues of research over the coming years.

## References

- Aksoy, S.; Tusk, C.; Koperski, K.; and Marchisio, G. 2003. Scene modeling and image mining with a visual grammar.
- Bryant, C.; Muggleton, S.; Page, C.; and Sternberg, M. 1999. Combining active learning with inductive logic programming to close the loop in machine learning. In *Proc. AISB Symposium on AI and Scientific Creativity*.
- Duda, R.; Hart, P.; and Stork, D. 2000. *Pattern Classification*. Wiley.

- Fern, A.; Givan, R.; and Siskind, J. 2002. Specific-to-general learning for temporal events with application to learning event definitions from video. *Journal of Artificial Intelligence Research* 17:379–449.
- Fitzpatrick, P.; Metta, G.; Natale, L.; Rao, S.; and Sandini, G. 2003. Learning about objects through action - initial steps towards artificial cognition. In *Proc. IEEE International Conference on Robotics and Automation*, volume 3, 3140–3145.
- Hargreaves-Heap, S., and Varoufakis, Y. 1995. *Game Theory, A Critical Introduction*. Routledge.
- Harnad, S. 1990. The symbol grounding problem. *Physica D* 42:335–346.
- Ivanov, Y., and Bobick, A. 2000. Recognition of visual activities and interactions by stochastic parsing. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 22(8):852–872.
- Kazakov, D., and Dobnik, S. 2003. Inductive learning of lexical semantics with typed unification grammars. In *Oxford Working Papers in Linguistics, Philology, and Phonetics*.
- Magee, D. R.; Hogg, D. C.; and Cohn, A. G. 2003. Autonomous object learning using multiple feature clusterings in dynamic scenarios. Technical Report School of Computing Research Report 2003.15, University of Leeds, UK.
- Magee, D. R. 2004. Tracking multiple vehicles using foreground, background and motion models. *Image and Vision Computing* 20(8):581–594.
- Moore, D., and Essa, I. 2002. Recognizing multitasked activities from video using stochastic context-free grammar. In *Proc. AAAI National Conf. on AI*.
- Muggleton, S. 1995. Inverse entailment and Progol. *New Generation Computing, Special issue on Inductive Logic Programming* 13(3-4):245–286.
- Petkov, N. 1995. Image classification system based on cortical representations and unsupervised neural network learning. In *Proc. IEEE Workshop on Computer Architectures for Machine Perception*, 430–437.
- Santos, P.; Magee, D.; and Cohn, A. 2004. Looking for logic in vision,. In *Proc. Eleventh Workshop on Automated Reasoning*.
- Siskind, J. 2000. Visual event classification via force dynamics. In *Proc. AAAI National Conference on AI*, 149–155.
- Stamenov, M. I., and Gallese, V., eds. 2002. *Mirror Neurons and the Evolution of Brain and Language*. John Benjamins.
- Sternberg, M.; King, R.; Lewis, R.; and Muggleton, S. 1994. Application of machine learning to structural molecular biology. *Philosophical Transactions of the Royal Society B* 344:365–371.
- Strehl, A., and Ghosh, J. 2002. Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research* 3:583–617.