Learning Spatio-Temporal Patterns for Predicting Object Behaviour

Neil Sumpter † ‡ and Andrew J.Bulpitt †

†School of Computer Studies, University of Leeds, LS2 9JT, UK
‡ Silsoe Research Institute, Silsoe, Bedford MK45 4HS, UK
neils@scs.leeds.ac.uk

Abstract

Rule-based systems employed to model complex object behaviours, do not necessarily provide a realistic portrayal of true behaviour. To capture the real characteristics in a specific environment, a better model may be learnt from observation. This paper presents a novel approach to learning long-term spatio-temporal patterns of objects in image sequences, using a neural network paradigm to predict future behaviour. The results demonstrate the application of our approach to the problem of predicting animal behaviour in response to a predator.

1 Introduction

The recognition of spatio-temporal patterns within a scene is an important facet of computer vision research. Future behaviour of an object, in terms of its motion and appearance, can be implied through a learned model of previous behaviour.

Short-term predicitions of likely object motion and deformation over one time-step allow objects to be tracked robustly through a scene. This has been achieved successfully using a Kalman filter framework [1], for example in fitting a deformable model of human shape to an image in order to track over image sequences [2]. Whilst such a framework is suitable for simple situations where the probability density function (pdf) of the input data can be represented as a single Gaussian distribution, it cannot support more than one simultaneous hypothesis. An alternative approach is that of the CONDENSATION algorithm [3], which represents the pdf as a number of sample hypotheses which can be stochastically propagated over time. This algorithm has been applied in a number of scenarios, for example tracking hand deformations through the use of a learned Markov chain [4].

Longer-term models of object behaviour provide descriptions of the higher-level processes occuring within a scene, for example the way in which objects interact with each other. Such interactions can be modelled by applying a set of rules which the objects must obey [5], but for intelligent living objects such as animals and humans the behaviours rarely simplify into explicitly stated laws. For example, the motion of a ball bouncing obeys physical laws of gravity, whereas a human's path can depend on the number of obstacles in their way or perhaps the familiarity of the route, which cannot be easily modelled as a set of rules. A more appropriate method would involve creating automatically a

BMVC 1998 doi:10.5244/C.12.65

model of object behaviour from observation, which would inevitably capture the inherent real-world properties of the scene. Such a method is used by Johnson and Hogg [6] to model human paths through a scene in order to recognise typical and atypical events.

In this paper, we present a new approach to the task of learning long-term spatio-temporal patterns of object behaviour. In doing so, we describe how the resulting model can be used to predict the location and shape of an object over time.

We detail a simple neural network architecture that quantises partial trajectories in location/shape-space. By applying a feedback mechanism, a probability density function of relationships between trajectories can be constructed, and used to choose the most likely future events depending on recently observed history.

Furthermore, we discuss how our approach can be used to model the interactions between objects, and how the knowledge of the behaviour of one object can be used to predict the future behaviour of another. A suitable application is discussed, where a model of animal behaviour in response to a predator is required. The results presented compare predicted paths of the animals with their actual behaviour.

2 Background

The application of neural networks to the task of temporal pattern recognition has been achieved with some success. The spatio-temporal motion of a walking pedestrian [7] was categorised by incorporating the previous five state vectors as inputs to a network, with each state vector representing the location and shape (as a spline contour) of the human at a given time. Whilst successful at tracking typical movements of the pedestrian, the extension to longer histories would imply a prohibitively large input vector.

Alternately, a system for recognising human motion was described in [8], where two ART networks were combined with a layer of decaying excitations, in order that longer sequences of movements of human body joints be categorised. Johnson and Hogg [6] describe a similar approach using vector quantisation to construct a pdf of human trajectories within a scene.

Our approach builds on the above, to allow the direct learning of spatio-temporal patterns such that the future behaviour of an object (or objects) can be predicted.

3 Modelling Object Behaviours

A network architecture is used as in Figure 1. Two competitive learning networks are connected by a layer of leaky integrators [9]. The two networks are labelled according to an anology with natural-language processing: the symbol network categorises the object shape and locations at any time (in other words, the characters - or symbols - observed), and the context network categorises the order in which they appear (the context of the characters). The layer of leaky integrators retains a history of previous events.

Typically our input vectors are of the order of 10-20 dimensions, depending on the number of objects being modelled and the number of parameters used to describe the shape variations.

Shape is modelled by a Point-Distribution Model (PDM) [10] in order to reduce the dimensionality of the shape data. For each object in a suitable training set, the object outline is represented as a B-spline curve with the curve's control points forming the

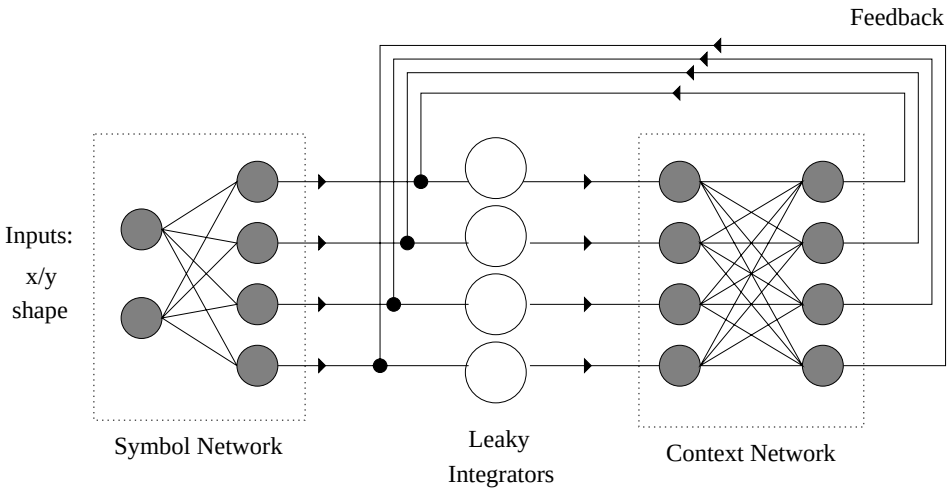

Figure 1: The approach used, represented as a network architecture

basis for the PDM. Each object is then represented by a mean shape and a small number of parameters describing the shape variance within the training set.

3.1 The Symbol Network

The purpose of the symbol network is to model the complex probability density function of an N-dimensional input vector at any given time. To achieve this, vector quantisation is implemented in the form of a competitive learning network [11].

Vector quantisation, as is true of most data-clustering techniques, is the representation of a feature space by a number of prototype vectors. In this case we model the pdf using M prototype vectors represented by the weights of the neurons in the competitive learning network. The weights are randomly initialised within the input feature space, and the following algorithm applied, over many epochs:

1. Let $\mathbf{x}$ be the input vector (normalised to be within the range [-1:1])
2. Find the winning neuron j at time t whose weights, $\mathbf{w_j^t}$ best represent $\mathbf{x}$
3. Update the weights as follows:

$$\mathbf{w_i^{t+1}} = \begin{cases} \mathbf{w_i^t} + \alpha^t(\mathbf{x} - \mathbf{w_i^t}) & i = j \\ \mathbf{w_i^t} & i \neq j \end{cases}$$

where α^t is the learning-rate, in the range [0,1], and decreases over time so convergence increases.

In the competitive learning network, the winning neuron is found by a suitable similarity measure, for example the Euclidean distance metric.

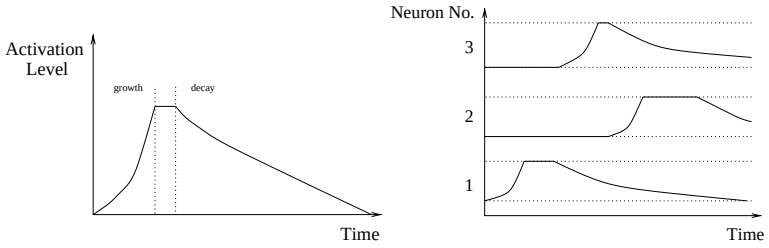

Figure 2: (a) A single integrator and (b) during training; neuron 1 starts off winning, then 3, and then neuron 2, by which time the activation on neuron 1 has decreased, but still holds a significant value.

3.2 Leaky Integrator Layer

We model the path an object follows through the input feature space by examining the activations it causes across the symbol network's output neurons. This is achieved by using a layer of leaky integrators connected to the output neurons.

A leaky integrator [9] is a neuron which stores a memory of its previous activation. The integrator will increase as the associated output neuron of the symbol network wins, and then decay as an alternate neuron wins (see Figure 2). At any one time, a number of integrators will have varying amounts of activation which represent the order in which the object's state in input feature space has changed.

The leaky integrators are implemented as follows:

$$a_i^{t+1} = \begin{cases} a_i^t + \beta a_i^t & \text{if neuron } i \text{ wins on symbol network} \\ a_i^t - \gamma a_i^t & \text{if neuron } i \text{ does not win} \end{cases}$$

where a_i^t is the activation level of neuron i at time t, and the values β and γ control the rate of growth and decay, and effectively govern the memory span of the neuron.

3.3 The Context Network

In order to model the pdf of object behaviours, we again use vector quantisation to place neuron weights within the vector space of the leaky integrator outputs, thus modelling the pdf of these activations. However, our approach differs to that of [6] by incorporating a feedback mechanism, from which we can learn the relationship between the activation trace at any time and the next input vector. In this way we are able to construct a model that can predict the next state from a given sequence of leaky neuron activations.

The feedback loop affects which leaky integrator wins, and subsequently which path of activations is selected. For this reason, the correct output node of the context network must win in order that the correct leaky activation sequence occurs. This requires a different learning algorithm than the normal competitive learning approach used in the symbol network:

1. Present input vector $\mathbf{x}$ to symbol network, and obtain output vector $\mathbf{O_x}$
2. Present current activation levels of leaky integrators, $\mathbf{A^t} = (a_1^t, \ldots, a_M^t)$, to the context network and obtain output vector $\mathbf{O_A}$

3. Combine (1) and (2) in weighted sum: $\mathbf{T} = \lambda \mathbf{O_x} + \mu \mathbf{O_A}$
4. Winning neuron i is the maximum valued component of $\mathbf{T}$
5. Force context network prototype i to learn current activation pattern $\mathbf{A^t}$ (using rule (3) in symbol network algorithm)
6. Update the leaky integrators using neuron i as the winner

During training the state of the symbol network will not necessarily change at each time step, and this can lead to the context network remaining stationary in one state. This is remedied by disallowing the context network from updating its weights (step (5) in the above algorithm), if the symbol network remains in the same state.

3.4 Prediction

When suitably trained, the network is inherently capable of prediction due to the recurrent nature of the feedback mechanism. This differs greatly from previous approaches. For instance, [6] requires an extra learning phase to link the partial trajectory outputs of a similar context network, achieved using a Markov chain [12].

The algorithm for prediction is similar to that for training the context network, except that no updating of the weights occurs. For long term prediction no input vector $\mathbf{x}$ is present, yet the model can continually predict using the recurrent nature of the feedback loop. The winning neuron is selected as before using only the context network. The new vector is recovered from the weights (on the symbol network) associated with this neuron.

It may be the case that part of the input vector is known, for example the location is known but not the shape, or vice-versa. The symbol network can then be used to provide a closest match to the partial inputs, and thus partially affect the predicted path of leaky activations. This proves advantageous with our application and enables the prediction of sequences with missing data.

4 Applying the Model

The method of predicting behaviours, as described in Section 3, provides a general framework for predicting any suitable vector in time. In our case, we are particularly interested in vectors that represent the state in terms of location and shape of an object, and now present an application which places importance on the prediction of object behaviours.

4.1 The Robotic Sheepdog Project

The Robotic Sheepdog Project [13], is an investigation into the nature of the interaction process between animals and machines. It aims to exploit these interactions such that an autonomous vehicle can be 'taught' to herd a flocking group of animals (in this case ducks) to a predefined goal.

In order for this to be achieved successfully, a model of the likely reaction of the animals to the robot vehicle must be constructed. Such a model can be built using a rule-based solution (based on [5]), which can be seen to be quite successful in providing a control strategy for the robot [14]. However, whilst this provides a simulation of animal

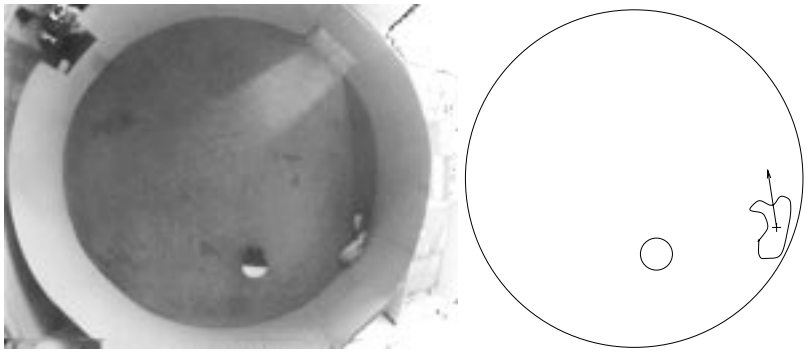

Figure 3: (a) A typical image of the arena, and (b) the corresponding input vector representation

flocking that is visually similar to real animal behaviours, it can be argued that a more appropriate model could be constructed automatically by observing the animals in their environment. For example, the animals prefer shadow-filled areas where they can 'hide' - a simulated model would not be able to distinguish such areas without extra rules, whereas the learned model would respond to such eccentricities.

The shape of the flock is also of interest, since it represents behavioural traits of the animals; for example, a long elliptical flock shape indicates panic as the animals flee from the robot predator. By including appropriate parameters in the input vector, flock shape is also predicted.

4.2 Data

In learning the behaviours of animals in response to the robot, we consider modelling the location of the robot and flock, together with the flock shape and velocity. The flock is modelled as a whole - not as individual birds due to the poor resolution of the image sequences (see Figure 3). The flock and robot move within a constrained environment - an arena, eight metres in diameter.

The flock is identified using background subtraction, and the outline of the resulting mass is used as the basis for a PDM, as described in [15]. Typically five shape parameters of the PDM are sufficient to describe at least 90% of the flock shape variation. The robot is segmented using a high-contrast black and white motif placed on top of the robot, and this is located within the image frame. This motif allows the robot's orientation to be deduced, as desired by the robot control strategy.

Figure 3 illustrates a typical image frame, and the corresponding information that we extract from each scene. The input vectors that represent each scene are of the form:

$$\mathbf{x} = (f_x, f_y, r_x, r_y, \partial f_x, \partial f_y, b_1, \ldots, b_5)$$

where $f_{x/y}$ is the flock location, $r_{x/y}$ is the robot location, $\partial f_{x/y}$ is the flock speed and b_i are the flock shape parameters.

We consider a set of 20 training sequences that represent typical behaviours of the animals. Each sequence consists of between 400 and 1200 frames and thus input vectors,

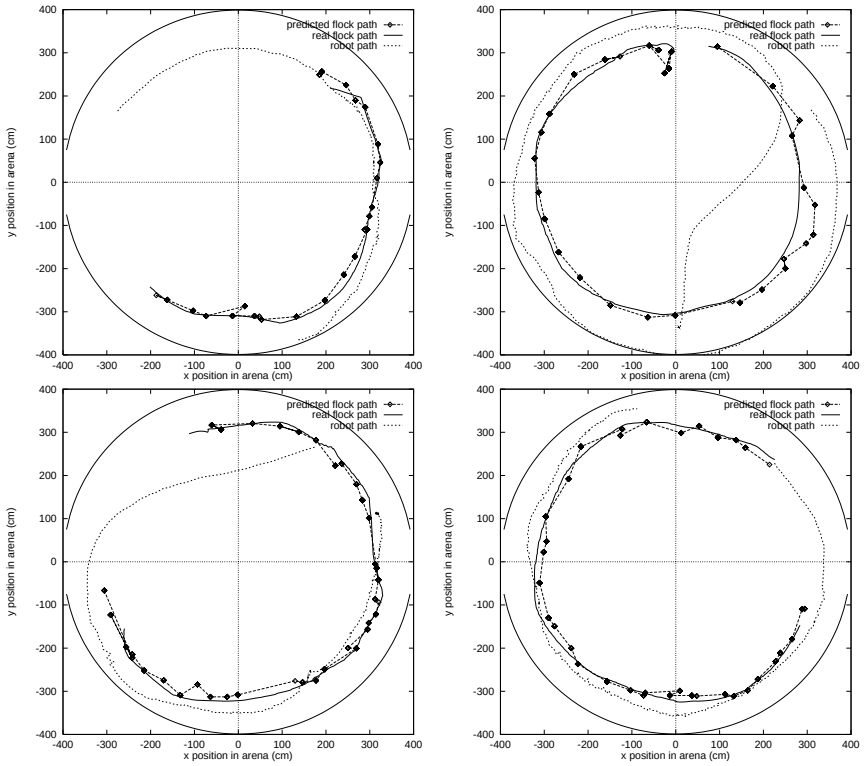

Figure 4: Results of prediction in comparison to actual flock path, (top) for sequences used to train the model and (bottom) for unseen paths

with each sequence beginning at different positions within the arena.

4.3 Results

The set of training sequences is presented to the network architecture. An input layer of 11 nodes (corresponding to the 11-dimensional vector $\mathbf{x}$ defined above) was used, and 500 output nodes for quantisation. For the symbol network, the order of input vectors is randomised to avoid 'dragging' the weight vectors through feature space, resulting in a more representative quantisation of the input vectors. The training algorithm iterates over 5000 epochs, with the learning rate α^t decreasing linearly from 0.4 to 0.0001 over the training period.

Once trained, we present each sequence to the model using the algorithm defined for the context network. The context network is trained for 1000 epochs, again with the learning rate decreasing linearly from 0.4 to 0.0001. The time constants of the leaky integrators are set as $\beta = 0.6$ and $\gamma = 0.06$, which effectively gives a memory of approx 20 input vectors.

Figure 4 shows typical results of prediction, using the trained model. The path of the robot from a sequence is presented, and the corresponding predicted path of the flock

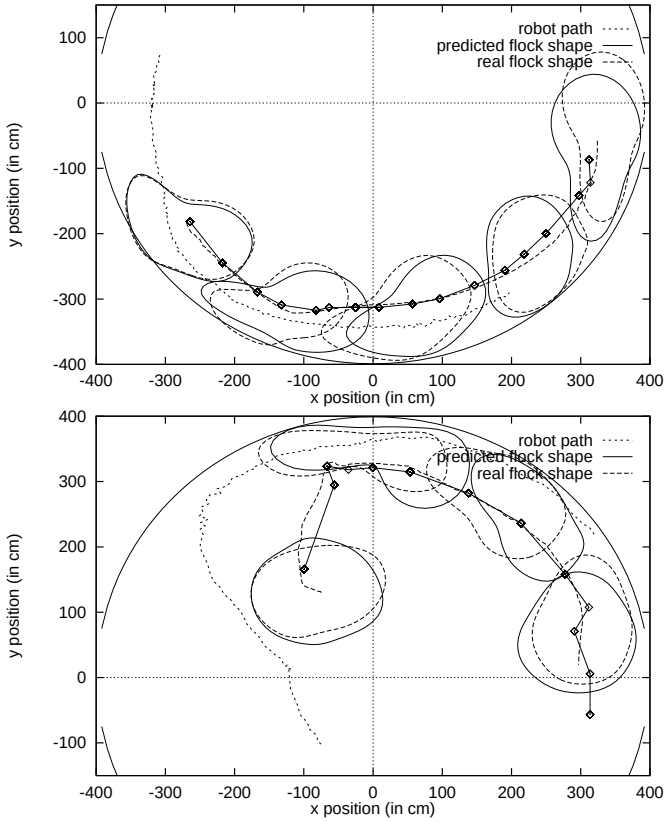

Figure 5: Shape Prediction; the flock shape is typically 'tadpole-like' with a large head of group where most animals flock, and a tail section consisting of the slower animals.

is shown. For both training sequences and unseen paths, it is observed that predicted behaviour closely represents the orginal path. In both instances, the flock path is only presented for the first 10 steps of the sequence in order to produce some initial history activations upon the leaky integrators.

Examples of the shape prediction can be seen in Figure 5. The shape of the flock is inherently poorly defined, due to restrictions in the image quality, but retains some interesting characteristics. The shape can be described as almost 'tadpole-like', where most of the animals are at the head of the group but some slower animals form a tail to the shape.

5 Conclusions

In this paper we have presented a method of learning the spatio-temporal patterns of objects. The derived model has an implicit method of prediction, which allows the extrapolation of future behaviours from recently observed histories. The model requires fewer

training steps than similar techniques, and provides a more appropriate model of object behaviour than methods which limit objects to move and deform according to predetermined rules.

We have described an application of this model, for predicting the trajectory and shape deformation responses of a flocking group of animals to a robot predator. Qualitative results have been presented that illustrate the accuracy of the model in the task of prediction of location and shape vectors from only partial input information. Current work in progress concerns producing a quantitive evaluation, providing a suitable measure of prediction accuracy.

Further work in using the model in our given application could lead to a complete control strategy. The model inherently predicts not only the path and shape of the flock, but also of the robot. It would therefore be quite feasible to carry out a limited search of possible animal paths that allow the animals to be herded to a goal, in order to extract robot paths that will maximise the likelihood of a successful autonomous herding.

6 Acknowledgements

The authors would like to thank Dr Roger Boyle at the University of Leeds, and Dr Robin Tillett at Silsoe Research Institute for their contributions to this work. This work is funded by an EPSRC grant, with additional support from BBSRC through Silsoe Research Institute.

References

[1] A. Blake, R. Curwen, and A. Zisserman. A framework for spatio-temporal control in the tracking of visual contours. *International Journal of Computer Vision*, 11(2):127–145, 1993.

[2] A.M. Baumberg and D.C. Hogg. An efficient method for contour tracking using active shape models. In *Proc. IEEE Workshop on Motion of Non-rigid and Ariculated Objects*, pages 194–199, Austin, Texas, November 1994.

[3] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *European Conf. of Computer Vision*, pages 343–356, 1996.

[4] T. Heap and D.C. Hogg. Wormholes in shape space: Tracking through discontinuous changes in shape. In *Proc. Sixth International Conference on Computer Vision*, pages 344–349, Bombay, India, 1998.

[5] C.W. Reynolds. Flocks, herds and schools: A distributed behavioural model. *Computer Graphics*, 21(4):25–34, July 1987.

[6] N. Johnson and D.C. Hogg. Learning the distribution of object trajectories for event recognition. *Image and Vision Computing*, 14(8):609–615, 1996.

[7] Li-Qun Xu and D.C. Hogg. Neural networks in human motion tracking - an experimental study. *Image and Vision Computing*, 15:607–615, 1997.

[8] A.J. Bulpitt. *A Multiple Adaptive Resonance Theory Architecture Applied to Motion Recognition Tasks*. PhD thesis, Dept of Electronics, University of York, 1994.

[9] M. Reiss and J.G. Taylor. Storing temporal sequences. *Neural Networks*, 4:773–787, 1991.

[10] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham. Training models of shape from sets of examples. In *Proceedings British Machine Vision Conference*, pages 9–18, 1992.

[11] D.E. Rumelhart and D. Zipser. Feature discovery by competitive learning. *Cognitive Science*, pages 75–112, 1985.

[12] N. Johnson, A. Galata, and D.C. Hogg. The acquisition and use of interaction behaviour models. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 1998. To appear.

[13] R. Vaughan, J. Henderson, and N. Sumpter. Introducing the robot sheepdog project. In *Proc. Int. Workshop on Robotics and Automated Machinery for Bio-Productions*, 1997.

[14] R. Vaughan, N. Sumpter, A. Frost, and S. Cameron. Robot sheepdog project achieves automatic flock control. In *Proc. Fifth International Conference on the Simulation of Adaptive Behaviour*, 1998. To appear.

[15] N. Sumpter, R.D. Boyle, and R.D. Tillett. Modelling collective animal behaviour using extended point distribution models. In *Proc. British Machine Vision Conference*, pages 242–251, 1997.