Joint spatial and temporal structure learning for task based control

Kingsley Sage and Hilary Buxton, University of Sussex
Department of Informatics / Centre for Research in Cognitive Science, Brighton, BN1 9QH, UK
{khs20,hilaryb}@sussex.ac.uk

Abstract

We present an example of a joint spatial and temporal task learning algorithm that results in a generative model that has applications for on-line visual control. We review work on learning transformed mixture of gaussians (due to Frey and Jojic) and Variable Length Markov Models (VLMMS due to Ron, Singer and Tishby). We show how a temporal model, learned through an extension of VLMMs to deal with multinomially distributed input symbol vectors, can be used as an improvement on Maximum Likelihood (ML) for prior parameter estimation for the Expectation Maximisation (EM) process.

1. Introduction

There is much current research interest in the development of frameworks for cognitive vision systems. Many of these frameworks draw inspiration from work on active (purposive) vision [1], [2], [3], [14], [15]. Task based visual control is one useful element of the cognitive vision paradigm and is concerned with the application of a task relevant control structure to guide low level vision processes. The benefit of this approach is computational efficiency both in terms of the low level processes and subsequent scene interpretation. Processing is limited to only that data that is considered task relevant. Task based visual control can therefore be thought of as data driven (bottom) up) processing limited in scope by task based (top-down) control.

It is important here to distinguish between knowledge based/driven extensions to classical (non-cognitive) approaches to vision and task based control. In the former, scene specific knowledge of object parameters (speed, size etc.) can be used to simplify scene interpretation or increase the effectiveness of tracking, but the vision task is still under-constrained with a corresponding high computational cost in solving correspondence issues. Relevancy is uniform across all parts of the scene. In contrast, task based control seeks to extend computational efficiency by limiting the processing to only those objects/parts of the scene that have task based relevance.

2 Review of field

Learning in task based visual control (and computer vision more generally) spans two separate (but related problems). On the one hand we need to learn the spatial structure of task relevant objects and, on the other, we need to learn the temporal dynamics (behaviour) of those task relevant objects. Classical approaches to computer vision have viewed these as separate problems (e.g. determination of spatial structure as a segmentation task, or analysing behaviour as a statistically time dependent process). Recent developments have included combining spatial and temporal learning into a single inter-dependent process (such as Frey and Jojic's transform invariant mixture of Gaussians, [6]) and the application of cognitive principles to exploit dependencies between the two modes of learning (Moore, Essa and Hayes [10]).

A further key aspect related specifically to the notion of control is that we need to build models that have generative properties. So for spatial structure we can generate pixel values (visualise structure prototypes) and for temporal dynamics we can use constraints on the step changes in the hidden variables to predict future observation data.

2.1 Learning spatial structure

If there is no prior information about the spatial structure then the problem is one of discovering a set of prototypes (a model λ) that best fit the training (observation) data O. I.e. we need to maximise $P(\lambda|O)$. There is usually no computationally tractable globally optimal or closed form solution for this task and the problem falls to gradient ascent techniques such as Dempster's Expectation Maximisation (EM) approach [5] , genetic algorithms, random search and so on (typically, any method that uses a likelihood function to model $L(O|\lambda)$). Additional variables may be incorporated into the EM learning. Frey and Jojic [6] use transformation indices (i.e. cluster transformation pairings) to create transform invariant mixture of Gaussian models. Further work by Frey and Jojic [7] showed how this process could be implemented computationally efficiently for all possible X and Y translations by modelling transformation as a convolution of the Fourier transforms of both the prototypes and O. Other types of invariance can then be achieved through appropriate co-ordinate system transformations (e.g. the log-polar form for arbitrary scale and rotations).

2.2 Learning temporal dynamics

Markov models have been used extensively in the modelling and recognition of human activities involving highly structure and semantically rich behaviours such as sign language (Starner and Pentland [13], and Vogler and Metaxis [16]), dance and aerobics (Galata and Hogg [8]) and vehicle movements around parked aircraft. Learning of first order Markov models is a well-established field. Unfortunately, as Galata and Hogg point out [8], Hidden Markov Models (HMMs) do not encode high order temporal dependencies easily. Local optima are frequently encountered by iterative optimisation techniques when learning HMMs with many free parameters. They use Variable Length Markov Models (VLMMs) [12] which are able to locally optimise memory length within the model. This approach captures long term temporal dependencies in some parts of behaviour and short term dependencies elsewhere.

Johnson and Hogg [9] model highly non-linear behaviours for a pedestrian tracking activity using a Gaussian mixture over system state change and observed history. Their approach is unusual in that it does not assume a discretisation of state space and models behaviours as both continuous and non-linear.

Up to this point in the discussion, there has been an implicit assumption that each spatial model (prototype) has its own independent model of temporal dynamics. Once again, this may under-estimate one's prior knowledge about what makes a good temporal model. Pal and Frey [11] developed their probabilistic montage based upon the idea that an image can be represented as a montage using many small transformed and cropped patches from a collection of latent images. Brand [4] show how coupled Markov models can account for inter-relationships between otherwise independent models.

2.3 Joint learning of spatial & temporal dynamics

Many researchers have separated the learning of spatial and temporal structure for vision applications. Frey and Jojic demonstrated that the two modes can be usefully unified in their transformed HMMs. Their approach combines the learning of transform invariant Gaussian mixture models with Baum Welch re-estimation. The HMM parameters provide an informed prior to the prototype/transformation index that enables the mixture modelling to reach a more optimal solution than with a uniform prior distribution (i.e. ML) alone. We will now show how variable length Markov models can be combined with the Transform Mixture of Gaussian (TMG) models.

3 Example visual task

To demonstrate the various approaches we use a simple visual task summarised in Figure 1 inspired by Frey and Jojic's pacman toy domain. In this task a 5x5 pacman shape follows a path defined by a cross on 25x25 grid. This path consists of 56 steps and can be summarised by a chain code. Each step of the path defines a 25x25 image with the appropriate pacman shape (always facing in the direction of travel) superimposed on a uniformly distributed noise background. The image sequence for the task then consists of 10 serial repetitions of the sequence giving a total of 560 frames. We constructed this task so that it contains higher order (and variable order) Markov temporal dependencies.

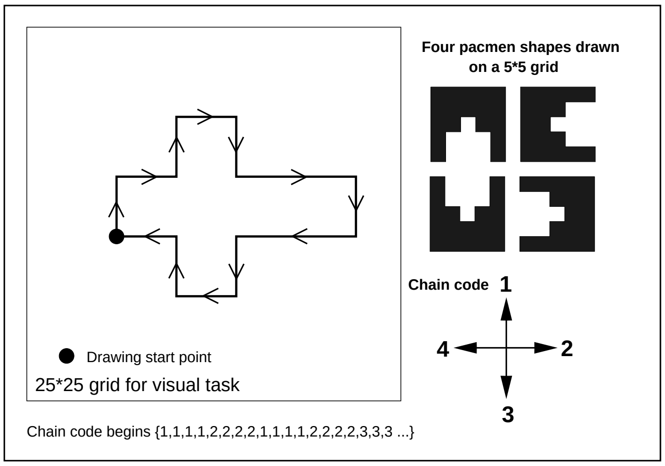


Figure 1. Summary of simple visual task

4 Applying TMG to the example visual task

The TMG method learns a mixture of gaussians model for a number C of different visual cluster prototypes normalised over a set of spatial transformations. Each cluster $c \in C$ has a latent image z_c formed by mean (μ) and covariance (Φ)parameters:

$$p(z_c) = p(z|c) = \mathcal{N}(z; \mu_c, \Phi_c) \quad (1)$$

The joint distribution of a training set image $x_t \in \mathbf{X}$, a transform l drawn from a set of L spatial transformation matrices $\Gamma = l_1, ...l_L$ and the latent image corresponding to cluster c with post-transformation noise Ψcan be defined as:

$$p(x_t|l, z_c) = \mathcal{N}(x_t; \Gamma_l z, \Psi) \quad (2)$$

$p(x_t)$ can then be defined by marginalising the expression for $p(x_t|l, z_c)$ over all $z_{c \in C}$ and Γ. Frey and Jojic then cast the learning of the cluster parameters μ_c and Φ_c as well as a post-transformation noise parameter Ψ as an Expectation Maximisation (EM) problem by setting the derivative of log $p(x)$ wrt the model parameters equals 0 and deriving the corresponding E and M step update equations. Frey and Jojic have produced an efficient implementation of this EM process for all possible $X \times Y$ translations using a convolution of the Fourier transform of the clusters with the training data. The source code for this implementation can be found at `http://www.psi.toronto.edu/~frey/tmgEM.m`. Other types of transformations can be accounted for by pre-processing the training set data by an appropriate co-ordinate transformation, such as the log polar transform for dealing with scale and rotation. For multiple transformations, such as X and Y translation and scale and rotation simultaneously, Frey and Jojic further developed a variational technique for decoupling transformation sets to reduce the computational complexity associated with multiple transformations.

5 Applying VLMMs to the example task

5.1 Integrating spatial and temporal learning

Frey and Jojic's original published MATLAB code is concerned with learning spatial structure only. Starting with the assumption that $P(C, L) = 1/CL$ for all timesteps T (uniform prior) and random values for the mean and variance values of the C cluster prototypes, it computes a post transformation probability estimate term $P(C)$ and then uses a ML estimate derived from that to inform the $P(C, L)$ prior at the next iteration ($P(c)/L$). The prior is assumed to be uniform across transformations i.e. $p(c, L) = constant$.

We first modified this code so that the $P(C, L)$ prior could be re-estimated independently for each frame so that we could use temporal models to constrain the prior estimates over time giving us $P(C_t, L_t)$. Frey and Jojic used this idea to build a first order Markov model onto their original work. The distribution of $P(C_t, L_t)$ depends not only on the individual post transformation probability $P(C_t)$, but on $P(C_{t-N}, L_{t-N})$ for an N-th order Markov process. Having computed $P(C_t)$ for each frame independently, we were then able to use variable length Markov modelling to determine $P(C_t, L_t)$ for the next iteration. This has the effect of improving the cluster prototypes by ensuring that they are estimated from the members of the training set that it is believed correspond to that cluster structure. Each training image contributes to the post transformation probability estimate term according to how likely it was based on the previous N training set images.

5.2 Overview of VLMM

Ron et al's formulation of the VLMM is based on optimisation of the statistical predication of a Markov Model measure by the instantaneous Kullback-Liebler (KL) divergence of the following symbols (the statistical surprise of the model when presented with the next symbol). The memory is extended when such a surprise is significant until the overall statistical prediction of the model is sufficiently good. The original formulation worked with training sequences that were strings of discrete symbols (such as in language modelling). The training process leads to a prediction suffix tree that predicts the probability that a symbol $s_t \in \Sigma$ follows a variable length string $s_{t-N}, ...s_{t-1}$. For our application Σ corresponds to the cluster indices and we model timestep to timestep cluster indices. More sophisticated modelling would expand Σ to model the cluster index plus other parameters such as relative motion, but there is a trade-off between improving the spatial description of the derived clusters and the additional computational complexity of the VLMM.

The central element in generating the prediction suffix tree is the KL divergence measure for the statistical surprise associated with a prediction for a symbol s compared with the longer string σs:

$$Err(\sigma s, s) = \sum\nolimits_{\sigma' \in \Sigma} p(\sigma s \sigma') log \frac{p(\sigma s \sigma')}{p(\sigma' | s) p(\sigma s)} \tag{3}$$

The source probabilities $p(s)$ and $p(\sigma|s)$ are estimated from the empirical counts (#) of their appearances in the post transformation probability estimates $P(C_t)$ according to Laplace's rule of succession:

$$p(s) \approx \frac{\#s + 1}{\sum\nolimits_{s' \in \sum^{|s|}} \#s' + |\Sigma|} \tag{4}$$

$$p(\sigma|s) \approx \frac{\sigma|s + 1}{\sum\nolimits_{\sigma' \in \Sigma} \#\sigma'|s + |\Sigma|} \tag{5}$$

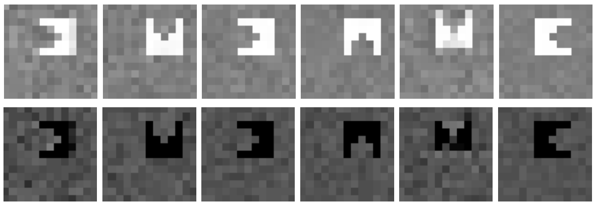
As $P(C_t)$ is multinomially distributed (it is a continuous valued vector of length C) rather than a discrete list of length T, the $\#s$ term becomes a probability value $P(c'_t \in C)$ rather than a count and the $\#\sigma|s$ becomes a product term:

$$\#\sigma|s \equiv p(c_{t-N}, ..., c_{t-1}|p(c_t)) = p(c_t) \prod_{t'=t-|N|}^{t'=t-1} p(s_{t'}) \tag{6}$$

This extension to the VLMM formulation enables us to estimate $p(s)$ and $p(\sigma|s)$ under noisy conditions and proves to degrade gracefully towards a first order (length=1) Markov model with $1/C$ state transitions with high levels of noise (e.g. during the first few iterations of the combined learning process).

The resulting prediction suffix tree is then converted into a Probabilistic Finite Automaton (PFA) 5-tuple (Q,Σ,τ,γ,π) where Q is the finite set of M variable length states derived (approximately) from the leaves of the prediction suffix tree, Σ is an alphabet of size C (the cluster indices), $\tau : Q \times \Sigma \rightarrow Q$ is the transition function, $\gamma : Q \times \Sigma \rightarrow [0, 1]$ is the output probability function and $\pi : Q \rightarrow [0, 1]$ is the probability distribution over the starting states.

We the use the PFA to estimate $P(C_t, L_t)$ prior for the next iteration. To do this we generate a forward evaluation trellis matrix of size $C * T$ which specifies how likely each cluster is for each frame in the training image set. We first create a forward evaluation trellis of size $M * T$. The probability values $p(q_t)$ are calculated as follows:



(a)

(b)

Figure 2. (a) Mean and (b) variance maps

$$p(q_t) = \Sigma_{q' \in Q} p(q'_t) \tau(q'_{t-1}, q_t) \quad (7)$$

We assume a uniform $1/Q$ distribution for π and to prevent the trellis co-efficients geometrically tending towards ∞ we normalise each column of the trellis such that $\Sigma P_Q = 1$ for each timestep. This ensure each timestep carries equal weight in the subsequent prior. Finally, we form the $C * T$ trellis by adding rows of the $M * T$ trellis such that the terminal cluster symbol for each of variable length states is the same. We then form the new prior by dividing $P(C_t)$ by the total number of possible transformations L for each individual timestep (our method is only modelling the cluster temporal ordering at the moment).

$$p(c_t, l_t) = p(c_t)/L \quad (8)$$

During the initial few iterations, the VLMM generates C states (one for each cluster) and a first order Markov model with each state having transition probabilities $\approxeq 1/C$. As the learning process continues and spatial structure starts to emerge, typically one of the cluster prototypes will acquire a value $p(c_t)$ significantly greater than the $1/C$ value (until the spatial learning has fully specialised the learned structure). This causes an attractor effect in the VLMM learning as the statistically most likely explanation for the temporal ordering will be a first order VLMM with the transition probability from the attractor state to itself for all timesteps. To overcome this attractor, we normalise the prior $P(C, L, T)$ prior to each EM iteration such that $\Sigma_T P(C_t, L_t) = k$.

5.3 Results

We trained our combined algorithm for 100 iterations with 6 clusters, allowing the VLMM to extend up to a depth of 5. The resulting mean and variance maps form the resulting 6 clusters are shown in Figure 2. Dark areas in the variance maps correspond to low variance.

The resulting VLMM prediction suffix tree consisted of 253 nodes (including the root node 'e') and the PFA had 211 nodes of length either 2 or 3. The shorter length nodes tended to correspond to portions of the task concerned with 'going straight ahead' and the longer ones with 'turning corners'. Clusters 2,3,4, and 6 captured most of the spatial structure for the task.

6 Conclusions

We have shown that variable order temporal modelling can be applied to a joint spatial and temporal learning task. The resulting model exhibits both spatial and temporal generative properties that have applications in the learning of visual control strategies. Given the current state of the joint model, we may stochastically predict the next. We plan to extend our experiments to real world images and to demonstrate an on-line control application as well as investigate other methods for updating the prior and model trimming. Model trimming is especially important in ensuring that the solution found is the most naturally interpretable one.

References

[1] Y. Aloimonos, I. Weiss, and A. Bandopadhay. Active vision. In *ICCV*, pages 35–54, London, UK, 1987.

[2] R. Bacsy. Active perception versus passive perception. In *IEEE Workshop on computer vision*, Washington, US, 1985.

[3] D. Ballard. Animate vision. *AI*, 41:57–86, 1991.

[4] M. Brand, N. Oliver, and A. Pentland. Coupled hidden Markov models for complex action recognition. *IEEE CVPR*, 1996.

[5] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the RSS*, 39:1–38, 1977.

[6] B. Frey and N. Jojic. Estimating mixture models of images and inferring spatial transformations using the EM algorithm. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 416–422, 1999.

[7] B. Frey and N. Jojic. Fast, large-scale transformation-invariant clustering, 2002.

[8] A. Galata, N. Johnson, and D. Hogg. Learning variable length Markov models of behaviour, 2001.

[9] N. Johnson and D. Hogg. Representation and synthesis of behaviour using gaussian mixtures. In *Image and Vision Computing 20(12)*, pages 889–894, 2002.

[10] D. J. Moore, I. A. Essa, and M. H. Hayes. Exploiting human actions and object context for recognition tasks. In *ICCV (1)*, pages 80–86, 1999.

[11] C. Pal, B. Frey, and N. Jojic. Learning montages of transformed latent images as representations of objects that change in appearance. *ECCV*, LNCS 2353:715–731, 2002.

[12] D. Ron, Y. Singer, and N. Tishby. The power of amnesia. In J. D. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems*, volume 6, pages 176–183. Morgan Kaufmann Publishers, Inc., 1994.

[13] T. Starner and A. Pentland. Visual recognition of American sign language using hidden Markov models. In *International Workshop on automatic face and gesture recognition*, pages 189–194, 1995.

[14] J. Tsotsos. On the relative complexity of active vs. passive visual search. *IJCV*, 79(2):127–142, 1992.

[15] S. Ullman. Visual routines. *Cognition*, 18:97–159, 1984.

[16] C. Vogler and D. Metaxas. ASL recognition based on a coupling between HMMs and 3D motion analysis. In *ICCV*, pages 363–369, 1998.