An information-theoretical approach to unsupervized learning over real-world data streams

François Chollet *, Yasuo Kuniyoshi

Intelligent Systems and Informatics Laboratory, The University of Tokyo
* fchollet@isi.imi.i.u-tokyo.ac.jp

September 27, 2012

Abstract

This paper proposes a new approach to the problem of inference and unsupervized learning in large, noisy data streams, based on the mapping of correlations between stream variables and leveraging the tools of Information Theory. Our approach was designed specifically for the needs of cognitive developmental robotics, such as online concept extraction from a video feed or other real-world sensor streams. We present experimental results demonstrating one-shot unsupervized sequencing and recognition of hand gestures videos displaying two timescales.

Introduction

MAchine Inference research so far has been concentrating on static spatial inference, and there are comparatively few inference models that operate over continuous data streams (spatiotemporal inference), such as video input, or the human sensorimotor stream. With the exception of Slow Feature Analysis (SFA) [1], the currently available dynamical models have usually been based on first performing static spatial inference over each frame of the temporal input, and then learning sequences over the spatial output. A notable example of this approach would be the Hierachical Temporal Memory (HTM) Zeta 1 model [2] by Dileep and Hawkins. Other examples include Hidden Markov Model (HMM)-based algorithms [3].

However, we argue that the inference paradigm illustrated by HTM and others is not adapated to cognitive robotics, and more generally, to massively dimensional, noisy real-world data streams (such as videos). Indeed, performing static inference over single frames maximizes the impact of high-frequency noise over the inference process, increasing the probability of an error. Moreover, ordered sequentiality is not a very tractable signal for understanding the real world, where higher-level inference is often the result of identifying unordered bags of features (for instance, scrambling words in a sentence does not significantly impair reading), and where reversed causality is generally as important as strict causality (as an example, it is more evolutionary useful to be able to process the unordered association "predator appears when neighbors flee and conversely", than to be limited to the naturally occurring, ordered sequence "predator appears, then neighbors flee"). In fact, we humans seem relatively poorly equipped for memorizing ordered sequences, while we have no trouble *associatively* recalling items that we have seen in a same context.

Furthermore, recent neuropsychology research [4] has shown that object recognition in monkeys had more to do with inter-frame phenomena such as temporal contiguity, than with the single-frame appearance of objects. This observation makes inter-frame correlation appear to be a good candidate principle on which to build a spatiotemporal inference system.

For these reasons, we believe there is a need for a new, better approach to spatiotemporal inference, one that would better fit the needs of cognitive robotics.

1 The model

1.1 Overview

To answer this need, we propose an inference model based on the mapping of correlations between variables of a data stream (such as the sensorimotor space of an agent), built using the tools of Information Theory [**?**]. It is based on the following fundamental hypotheses:

- A continuous data stream (such as the a human's or a robot's sensorimotor stream) can be structured into successive "situations" displaying strong information structure coherence

- Over each such situation, the relevant spatiotemporal concepts that define the situation form clusters of highly intercorrelated data points

- The spatial configuration of these clusters constitute a characteristic signature that can then be used for concept learning and recognition.

The intuitive idea behind this model is to parse the sensorimotor space according to "situations" featuring clusters ("concepts") of events or objects that reliably appear together, where the notions of "situation" and "reliably appear together" are defined using Information Theory. For instance, a particular scene in a movie will be interpreted as a "situation", and a character appearing on screen during this scene will constitute a spatiotemporal concept of the situation (since it exists as inter-correlated pixel values on screen).

The high amount of generality provided by its foundation on Information Theory allows for this model to be used as the basis for not only sensory inference algorithms, but generally for all aspects of sensorimotor development such as prediction and motor control. We briefly discuss prediction in the 'Future work' section.

1.2 Modularity and hierarchy

Due to its excellent properties of complexity reduction for large spatiotemporal problem spaces, we chose to adopt the widely used modular-hiearchical architecture principles [5].

As any other modular-hierarchical model, our architecture is made of a number of identical modules (with scale-specific calibration), organized into several layers representing temporal scales (fig. 1). Each scale is a collection of concept-specific neurons. These scales evolve at characteristic times that are increasingly long ("increasing invariance"). The concepts represented in a given scale are thus increasingly temporally extended. The zerot-th scale is occupied by the raw sensorimotor space X, a large set of random variables xi taking values in [0,1] and updating their value at every timestep dt. Each module takes for input a segment of the output of the lower scale, and outputs a segment of the input of the next scale (fig. 1).

1.3 Function of a single module

Each single module (fig. 1) performs correlation-based spatiotemporal dimensionality reduction over its input. This is done in two steps:

- First, the input is temporally sequenced into situations featuring particularly strong variable inter-correlation

- Then for each such situation, a spatial cluster of the variables most intercorrelated is extracted and is fed into a "classical" spatial inference system.

The entire process is happening online, though not in real time (concepts only get processed after they are already no longer present in the input).

The role of the first step, situation sequencing, is to allow for the computation of "clean" spatial clusters. Arbitrary sequencing would results in spatial clusters overlapping different naturally occurring situations, which would thus be poorly defined clusters making later spatial inference difficult.

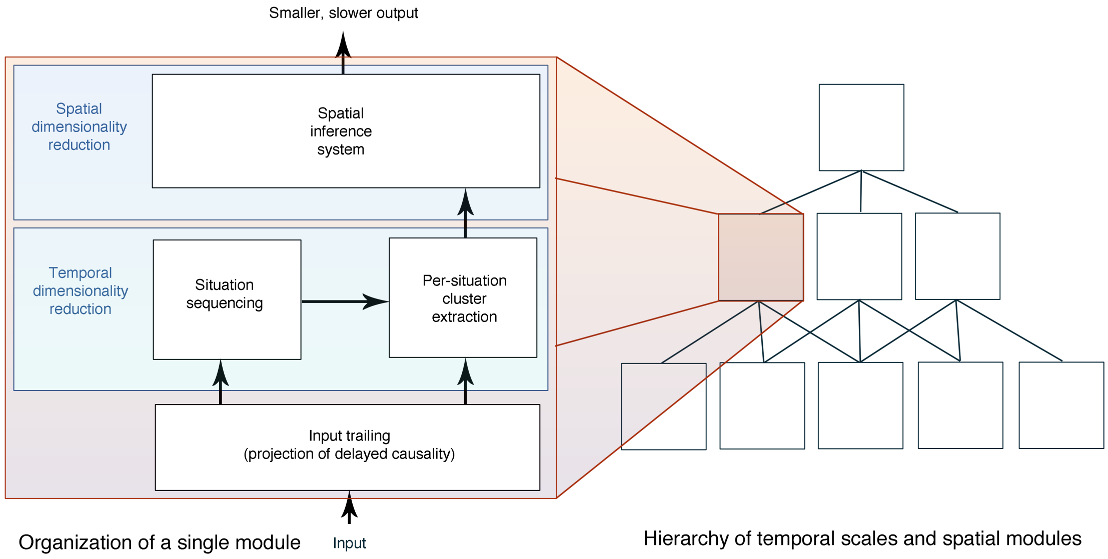
Smaller, slower output
Spatial dimensionality reduction
Spatial inference system
Temporal dimensionality reduction
Situation sequencing
Per-situation cluster extraction
Input trailing (projection of delayed causality)
Organization of a single module
Input
Hierarchy of temporal scales and spatial modules

Figure 1: Architecture overview

2 The algorithm

In this section we present the algorithmic detail of each separate element of a single module in our model. These are the algorithms used to produce the experimental results of section III.

2.1 Situation sequencing

Online situation sequencing is performed by computing the *quantity of structure* in the input, and identifying increases (start of a coherent situation) and decreases (end of situation) in this quantity. We chose the information-theoretic "Integration" as a measure of this quantity. Integration is a global estimate of the amount of statistical dependence within a set of variables $X = x_1..x_n$, defined as the difference between the sum of individual entropies on the unitary variables and the joint entropy of the entire set:

$$I(x) = \sum_i H(x_i) - H(X)$$

Where entropy H is defined as:

$$H(X) = -\sum p(x) log(p(x))$$

Intuitively, Integration increases with the decrease of joint entropy, ie. with the amount of statistical dependency within the set. It is highest for a set of highly correlated, high-entropy variables.

In our algorithm, pivots between situations in the input of a module are identified as:

- local minima of the average of integration pooled over all modules connected to the same "father" module, if the module has a father

- local minima of the integration value of the module input, if the module is the top module in the hierarchy

These local minima are taken over segments of time equal to the (pre-evaluated) characteristic length of a situation at the scale considered (thus the end of a situation and the start of a new situation is identified typically with a delay of half this duration). The intuitive idea here is that the apparition or disappearance of a concept in the input stream (such as a discontinuous camera move in a movie, or the apparition or disappearance of a character on screen) will result in sudden decreases of the quantity of structure in the information space of the stream. Segments of time between two such decreases will thus correspond to the presence of a single coherent concept in the stream.

The only module-specific calibration found in the model intervenes at this level: the expected characteristic length of a situation, ie. the length of the time segment over which an integration minimum will be identified as a situation pivot.

2.2 Note on computing information measures online

The computation of entropy, integration or mutual information for a random variable would normally require as a first step to establish an histogram of the values of the variable associated with their respective probabilities. However this method prevents online computation: the entire signal need to be available before processing can start. Let us consider the following instead: if we are dealing with variables at values in $[0, 1]$, then we can consider the value of the variable over each time step to be the probability of activation of a *prior binary variable* evolving at a much faster time scale. For instance $x(t) = 0.24$ would mean that a measure of the value of the prior binary variable b_x associated to x, at a random time between t and $t + dt$, would have a 24% probability of yielding 1. By working in the space of these prior binary variables b_x rather than the space of continuous variables x, we can assimilate values of x to probabilities of b_x, and thus bring information-theoretical computation online.

2.3 Cluster extraction

Once the start of a situation has been identified, the module computes at each time step the matrix of pairwise Mutual Informations of its input variables since the start of the situation. Mutual Information is the most general measure of correlation between two random variables, similar to Pearson's correlation coefficient, but able to capture non-linear dependencies. It is defined as:

$$MI(x, y) = p(xy) log \frac{p(xy)}{p(x)p(y)))}$$

The pairwise Mutual Information matrix can be intuitively interpreted as the spatial map of correlations within the input over the situation considered. It depicts clusters of particularly correlated variables, ie. spatiotemporal concepts. The matrix is symmetric positive and can be diagonalized in $\mathbb{R}$. Each of its eigenvectors characterizes one such independent concept.

It is to note that the use of mutual information here makes the inference process structurally very noise-resistant. Any co-occurence event that does not happen with statistical significance over the situation simply will not be processed.

2.4 Note on delayed causality detection

Mutual information cannot grasp delayed causality (nor does Integration), whereas almost all causality effects in real world data come with some delay. A simple solution to force the processing of delayed causality would be to compute not only the map of $MI(x_i(t), x_j(t))$ but also the maps of $MI(x_i(t), x_j(t + dt))$ with dt in a certain discreet interval. However the process would be computationally expensive and severely unelegant. The approach we have chosen instead is to give an immediate spatial representation to delayed causality through the use of an input pre-processing "trailing function":

$$X(t) = max(X(t), f.X(t - 1))$$

Where $0 < f < 1$ is the trailing factor.

Intuitively, this trailing function will force successive variable spike events to partially overlap, leading to a higher mutual information between the variables that saw these spikes. All input is pre-processed as such before entering a module of our architecture. The trailing function features a calibration parameter that determines the temporal sensitivity or causality detection. This parameter is linked to the only module-specific calibration parameter: the characteristic length of a situation at the scale considered.

2.5 Inference

The situation eigenvectors previously computed can then be fed into a "classical" spatial inference system, that will perform spatial dimensionality reduction (projecting the n-variable vector into a comparatively small space of expected concepts). Good candidates for such a system would be a sparse autoencoder [6] or a Kohonen self-organizing map [7]. However these systems require large training sample sets, whereas we would like to be able to perform few-shots learning. Therefore, rather than an algorithm with a learning phase, we use a simple locality-sensitive hashing function. It transforms the large input vector into a much smaller vector while roughly preserving space geometry.

Let n be the size of input I and m be the chosen size of the output. We chose a set $S = V_i, 0 < i < m$ of m normalized (for $L2$) random vectors with positive coordinates. The output is the vector of components $x_i = I.V_i$.

The output of the module is the output of the chosen learning and recognition algorithm (or in our particular implementation, the output of the hashing function). It is spatially much smaller than the input, and evolves much slower (it will be quasi-constant over each situation).

3 Experimental results

The above algorithm can be applied to extract the spatiotemporal concept hierarchy of any large continuous data stream.

In order to give a concrete illustration of how the algorithm works, we present here results from running it on a 27 second long, 30x40px grayscale video of continuous hand gestures (video described in fig. 2). Our particular instance of the model had two levels, the first one featuring 35 modules of 100 variables (nearly 200% global redundancy over the original input) and the second one constituted by a single module taking a 350 variables input (10 variables per child module).

The video input consists of 5 gestures (3-5 sec. each) coming in the sequence A - B - C - B - A (fig. 2). Each single gesture is made of several micro-gestures (eg. single hand wave) of about 1 sec. each, repeated 3 to 4 times. The input thus displays two time scales: the scale of micro-gestures and the scale of gestures. The second instances of gestures A and B differ slightly from the first instances in hand position, gesture duration, etc.

Experimentally, our algorithm is able to distinguish between the tree different unique gestures, and recognize both instances of A and both instances of B being the same gesture.

At level 1, the modules identify 21 micro-situations, the timing of which indicates they correspond to the micro-gestures in the video. The average over all first-level modules of the pairwise proximity between the eigenvectors representing these situations is shown in figure 2, already making apparent the global structure of the video.

At level 2, the single top module identifies 5 situations, corresponding to the 5 different hand gestures. Pairwise proximity between the eigenvectors of these situations is shown in figure 2, showing strong perceived proximity between the two instances of gesture A and between the two instances of gesture B.

It is to note that these results are obtained online, but not in real time: each situation gets "perceived" after it has already ended (with a delay of half the characteristic time of situations at the level considered). However we believe the algorithm can be ported to quasi real-time, which will be part of future improvements.

4 Future work

4.1 Real time processing

Our algorithm would be a good candidate inference system to build an embodied develop-

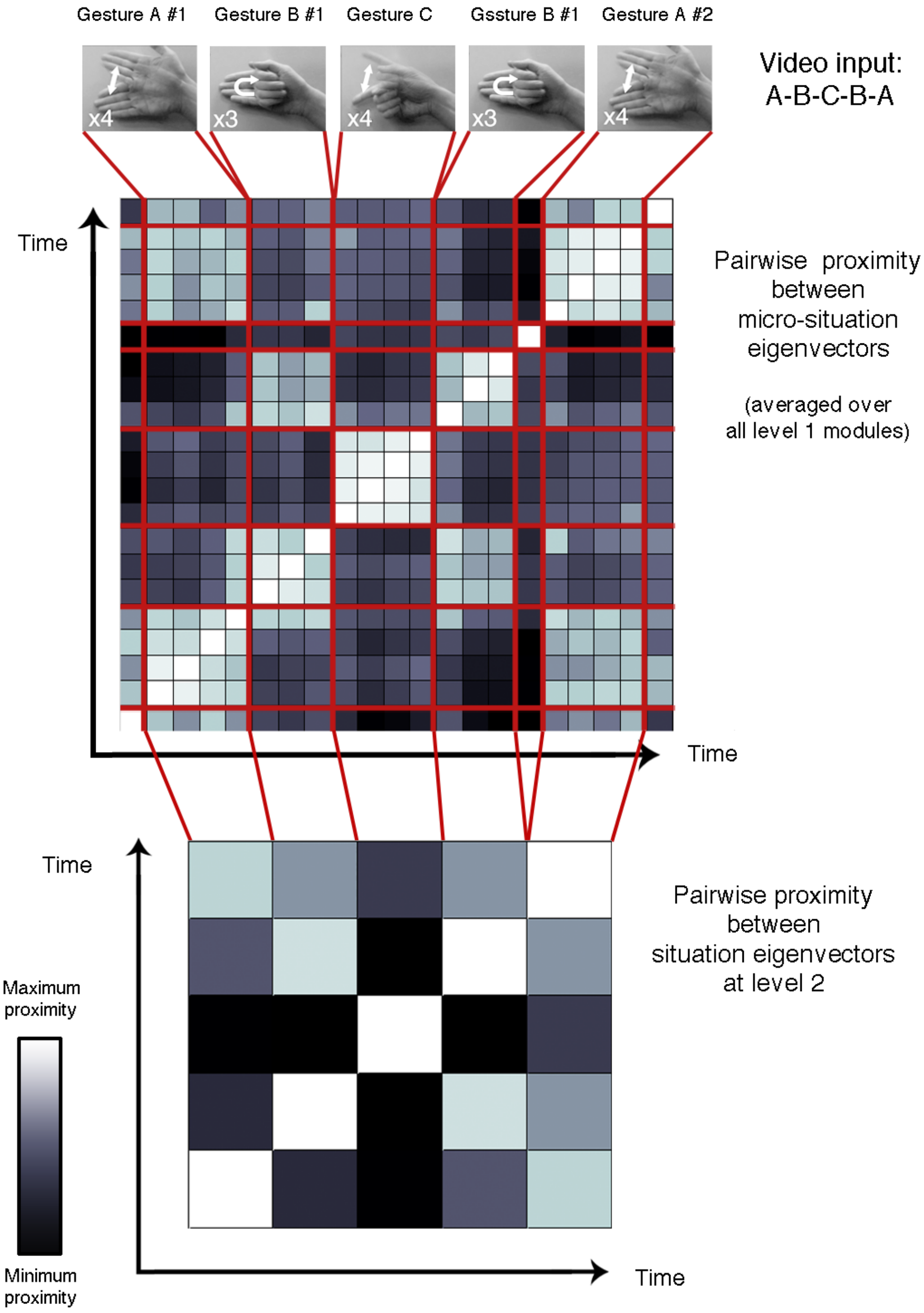

Figure 2: Learning the structure of a video with two time scales

mental robot –were it able to work in real time. We believe we will soon be able to port it to quasi-real time, with perceptive delay being only a fraction of the characteristic length of a situation. Indeed, we have verified experimentally that situation eigenvectors computed for only the first few frames of a situation are fairly close to those of the entire situation (which makes perfect sense given that situations are defined as segments of time displaying high coherence).

We can therefore improve the algorithm by having two separate processes, a delayed learning process and a quasi-real-time inference process. The former will be based on eigenvectors computed retrospectively, once a situation has been recognized as over. The later will be based on eigenvectors computed as soon as the start of a situation has been detected (or possibly even earlier if we base them on an arbitrary sequencing), which would be compared to eigenvectors recorded during the learning phase, in order to adjust in real time the probably distribution over what situations the system might be currently seeing.

4.2 Prediction

Because the notion of correlation that we employ in our model contains some temporal extension, If the system is able to recognize a situation after perceiving only its first few frames, it can also "predict" what follows in the situation, by comparing the correlations it has detected so far to the correlations mapped in the previously learned situation. Basically, having a certain number of variables in a cluster being "activated" would be enough to identify the entire cluster as being currently perceived, leading to all variables in the cluster to be activated. This can be achieved with simple top-down, level-to-level feedback.

4.3 Benchmarking

While the present paper is meant merely as a presentation of our approach, our algorithm still requires to be benchmarked against other spatiotemporal inference systems (SFA, HTM, HMM-based algorithms, etc.), in particular in regard to the tasks with applications in cognitive robotics, such as inference over a video stream or other real-world sensor data stream.

Conclusion

In this paper we presented a new approach to spatiotemporal inference, built with the tools of information theory. Our algorithm based on this approach is able to extract the spatiotemporal concept hierarchy of a large continuous data stream such as a video, as illustrated in our experimental results. It is our hope that future work on this model will yield interesting cognitive developmental robotics applications, allowing real-world agents to make sense of large sensorimotor data streams: performing one shot learning of increasingly complex concepts, hierarchical inference, prediction, and motor control.

References

[1] Laurenz Wiskott and Terrence J. Sejnowski, *Slow Feature Analysis: Unsupervised Learning of Invariances*. Neural Computation 14, 715-770, 2002.

[2] Dileep George and Jeff Hawkins, *Towards a Mathematical Theory of Cortical Micro-circuits*. PLOS Computational Biology, 2009.

[3] Rabiner, L. and Juang, B., *An Introduction to Hidden Markov Models*. ASSP Magazine, IEEE, 1986.

[4] Nuo Li and James J. DiCarlo, *Unsupervised natural visual experience rapidly reshapes size invariant object representation in inferior temporal cortex*. Neuron 67, 1062-1075, 2010.

[5] Michael I. Jordan and Robert A. Jacobs, *Modular and hierarchical learning systems* . M. Arbib (Ed.), The Handbook of Brain Theory and Neural Networks: Second Edition, 2002.

[6] Hinton, G. E. and Salakhutdinov, R.R, *Reducing the dimensionality of data with neural networks*. Science, 2006.

[7] Teuvo Kohonen, *The self-organizing map*. Proceedings of the IEEE, 1990.