A recursive patterns matching model for the dynamic pattern recognition problem

Eduard Puerto, Jose Aguilar & Danilo Chávez

Taylor & Francis
Taylor & Francis Group

Check for updates

A recursive patterns matching model for the dynamic pattern recognition problem

Eduard Puerto [a], Jose Aguilar [b], and Danilo Chávez [c]

aUniversidad Francisco de Paula Santander, Cúcuta/Colombia; bUniversidad de Los Andes, Merida, Venezuela; cEscuela Politecnica Nacional, Quíto, Ecuador

ABSTRACT

This paper defines a new recursive pattern matching model based on the theory of the systemic functioning of the human brain, called pattern recognition theory of mind, in the context of the dynamic pattern recognition problem. Dynamic patterns are characterized by having properties that change in intervals of time, such as a pedestrian walking or a car running (the negation of a dynamic pattern is a static pattern). Novel contributions of this paper include: (1) Formally develop the concepts of dynamic and static pattern, (2) design a recursive pattern matching model, which exploits the idea of recursivity and time series in the recognition process, and the unbundling/integration of pattern to recognize, and (3) develop strategies of pattern matching from two major orientations: recognition of dynamic patterns oriented by characteristic, or oriented by perception. The model is instantiated in several cases, to analyze its performance.

Introduction

The science of the brain has as a general aim, know and understand how the brain processes determine the cognitive functions (perception, memory, intelligence, learning, reasoning, consciousness, etc.) (Kaku 2014), (Xu et al. 2015) (Moser and Moser 2014). The brain (more specifically, the neocortex) is continually trying to make sense of the input presented. If a level of the neocortex is unable to fully recognize a pattern, it is sent to the next lower level. If none of the level can recognize a pattern, it is deemed to be a new pattern.

There are several projects dedicated to study the human neocortex. For example, BRAIN (National Institutes of Health 2016) is a project that tries to create a wide-ranging picture of the brain activities, while HBP (Markram 2012) attempts to create a computational simulation of the human brain, among other projects. "On Intelligence" (Hawkins and Blakeslee 2007) and "How to Make a Mind" (Kurzweil 2012) are complementary theories about how the brain works, proposing a systematic description of the neocortex (the brain region responsible for the

cognitive functions of high level) for the pattern matching. These two theories describe a systematic model of the operation of the neocortex, on the basis that the process of pattern matching is carried out through modules of pattern matching.

On the other hand, pattern matching is a field of research which aims to describe and classify data, objects, or general patterns within a category or class (Liu et al. 2016), (Pavlidis 2013), (Watanabe 2014), (Aguilar and Colmenares 1997), (Aguilar 2004). Several models for pattern matching exist at the computational level: Rules Based Models (Bobrow 2014), Classic Machine Learning (Alpaydin 2014), and Representational Learning (LeCun et al. 2015).

In general, the pattern matching problem has been studied in the literature from different points of views. Some similar approaches to our proposal are: wavelet theory, which applies a set of filters, tuned to extract features of specific spatial frequencies and sizes in the original image (Taubman and Marcellin 2012). The difference of our proposal and the wavelet theory applied to recognize, is that it is based on the extraction of features intrinsic of the pattern (edges recognition, texture classification, etc.), while our model considers features more generals, which are broken down recursively.

Today, Deep Learning is a new area of machine learning research. Deep Learning allows the computer to build complex concepts out of simpler concepts. Currently are been built pattern matching approaches based on Deep Learning, such as Convolutional Neural Network (CNN) (LeCun, Bengio, and Hinton 2015), Deep Boltzmann Machine (DBM) (Srivastava et al. 2013), Deep Belief Networks (DBN) (Lopes and Ribeiro 2015), among others. The main difference between Deep Learning and our model is that it is a feedforward hierarchy model and our model is a hierarchical-recursive model.

In this paper, we propose a formal model of pattern matching based on these novel ideas about the operation of the brain, for the dynamic pattern matching. The approach is called, the recursive pattern matching model (AR2P, by its initials in Spanish). Our model differs from previous models in that it can recognize dynamic patterns, which has not been extensively studied. To recognize the dynamic patterns, we defined two theorems. The first theorem focuses on recognition oriented by characteristics, and the second theorem on recognition oriented by perception. Our formalism works based on thresholds and time series, and is embedded in a recursive hierarchical process dynamically. The rest of the paper is organized as follows. The following section describes the recursive pattern matching model (AR2P). Then, the next section describes the formulation of the problem of dynamic pattern matching. Finally, in the last section are presented case studies.

Recursive pattern matching model (AR2P)

The model exploits the idea of recursivity in the recognition process, or unbundling/integration of pattern to recognize. Each layer in the

hierarchy is an interpretation space (ovals) identified as X_i, from i = 1 to m. X_1 is the level of recognition of atomic patterns, and X_m is the level of recognition of complex patterns (a complex pattern is characterized by being composed of patterns of lower levels). Each level is composed of Γ_{ji} recognition modules, (for j = 1, 2, 3... # of modules at level i).

ρ_{ji} is the recognized pattern by the module j at level i. The function of each recognition module is to recognize its corresponding pattern (i.e., a copy between multiple copies of the patterns in the real world). $s()$ represents the presence of a pattern to be recognized. This input is specific to each recognition module. For the top-down case, the output signal of the higher-levels is the input signal at the lower-levels.

There is a ν relationship of structural composition among the Γ_{ji} of different X_i, such that $\Gamma_{rt} \rightarrow \Gamma_{lk}$, where t < k, and the relationship "$\rightarrow$" indicates that Γ_{rt} of X_t is contained or forms part of Γ_{lk}, which belongs to layer X_k of higher level. In other words, a Γ of X_k is composed of different Γ of X_t of lower level. There may be different versions of the same pattern (redundancy/robustness) represented by different Γ_{rt}, from r = 1,2,3... until possible variations of the object in the real world. Each level i produce an output signal (recognition or learning) based on the responses of its modules. The output of each Γ_{ji} consists of a specific signal of recognition of its pattern ρ_{ji}, which is transmitted through the dendrites to its higher levels. This signal contains information about the characteristics of the pattern that represents. This process is valid for both, top-down and bottom-up processes. Such recognition is diffused through all the dendrites of which the recognition module is connected. When it is not recognized, it sends a signal that maybe involves learning.

The model considers two events, which can produce a learning process: New Learning occurs when the input pattern is not recognized (there is not a module that recognizes it), and Reinforcement Learning occurs when the input pattern is recognized. The first case occurs when the atomic patterns are recognized, but there are not modules that recognize them as a whole. The result of this process is a new recognition module for an unknown pattern. The new module is located in the hierarchy, at a level above of the atomic patterns recognized. This learning process uses as source of learning, the recognized active signals. Its aim is to update the information in the modules (Puerto Cuadros and Aguilar Castro 2016a).

Descriptive formulation of the problem of dynamic patterns matching

This section describes the extension of the problem of static pattern matching (described in (Puerto and Aguilar 2016b)) toward the dynamic pattern matching. In (Felzenszwalb, McAllester, and Ramanan 2008) the analysis of dynamic patterns is made from three aspects: exploration/exploitation of the

representation, use of the principle of "divide to recognize", and use of intrinsic contextual information. Other studies have attempted to exploit more information about the pattern, for example, the deformable parts of the patterns (Kelso 2014), or the dynamic changes of the patterns.

In general, a dynamic pattern is a pattern that changes in characteristics or perception, in a time interval, where: (a) it is an abstraction of a spatial or temporal object; (b) it is a collection (possibly orderly and structured) of descriptors that represent it; (c) each descriptor is a characteristic derived from the pattern, which is useful for recognition, and is represented by a variable bounded in a finite domain of values; (d) the descriptors represent the set of semantic information of interest; (e) the change is the variability in the time of the descriptors (domain of possible values).

A dynamic pattern is recognized; when in the evaluation of the descriptors (both characteristics and perception) in a time interval $[t_k, t_j]$, changes are detected in the domain of possible values. In general, the recognition of dynamic patterns can be classified into two types:

- Dynamic patterns oriented to characteristics (henceforth **DpoC**): A pattern is DpoC when has characteristics that change over time, for example, the expression of the emotions.
- Dynamic patterns oriented to perceptions (henceforth **DpoP**): A pattern is DpoP when the perception of the pattern is going to change (seen as a whole: form, appearance), according to what we go viewing. For example: Moving vehicle.

Mathematical formulation of a dynamic pattern

A dynamic pattern is formally defined as a three-tuple.

$$\rho_d = \ < D_n, f_\Delta, \Delta\tau > \ , \text{where} \tag{1}$$

$$D_n = \left[d_1, d_2, d_3 \ldots d_i, d^1, \ d^2, d^3 \ldots d^j \right] \tag{2}$$

d_i denotes the characteristic descriptors and d^j denotes the perception descriptors. The perception corresponds with our senses: visual, auditory, etc. (or combination thereof). Each descriptor has a domain that is the universe of possible values. For example, numeric values, labels, continuous, etc.

$$\widehat{d_x} \equiv \left[v_{1x}, v_{2x}, v_{3x}, \ldots, v_{kx} \right] \tag{3}$$

$$\widehat{d^y} \equiv \left[v^{1y}, v^{2y}, v^{3y}, \ldots, v^{ky} \right] \tag{4}$$

$\widehat{d_x}$ is the domain vector of d_x (x characteristics) and has a ranging of values from v_{1x} to v_{kx}, with $x = 1,2,3\ldots$to an arbitrary value, depending on the domain. $\widehat{d^y}$ is the domain vector of d^y (y perceptions) and has a range of values from v^{1y} to v^{ky}, with $y = 1,2,3\ldots$to an arbitrary value, depending of the domain. k is the value of the descriptor at the time t. If the change is continuous, fcd_i fcd^j are the change functions to each descriptor. These functions are collected in f_Δ; f_Δ is a vector of functions. The cardinality of $|f_\Delta| = |D_n|$.

$$f_\Delta = \left[fcd_1, fcd_2 \ldots fcd_i, fcd^1, fcd^2 \ldots fcd^j \right] \tag{5}$$

$\Delta\tau = \Delta\tau d_i \mid \Delta\tau d^j$ is a vector of "change event" of each descriptor d_i, d^j in D_n. It is a time series for each descriptor, ordered chronologically (could be continuous). Each descriptor can have a different time rate of change. For example:

$$\Delta\tau d_1 = [(t_1, v_{11}), (t_2, v_{21}), \ (t_3, v_{31}) \ldots (t, v_1)] \tag{6}$$

In the ordered pair $(t, \ v_i)$; the first element represents the change time, and the second element the value obtained. If a descriptor remains constant, its exchange rate is equal to zero (0), $\Delta\tau d_i \mid \Delta\tau d^j = 0$.

Description of the module for a dynamic pattern matching

A module for the dynamic pattern matching contains the information needed to recognize a dynamic pattern (descriptors, descriptor states, dynamics, and weight). $\Gamma\rho_d$ notation is used to represent this module. A $\Gamma\rho_d$ is formally defined as a three-tuple:

$$\Gamma\rho_d = \langle E_d, U_d, S_o \rangle, \tag{7}$$

where E_d: is an array composed of two-tuple $E_d = \langle S_d, C_d \rangle$ (see Table 1).

S_d: $S_d = \langle$ Signal, State$\rangle$ is an array that represents the set of dynamic signals that conform to the pattern recognized by $\Gamma\rho_d$ and their respective states. C_d: $C_d = \langle$ P, W$\rangle$, P are pointers to each time series $\Delta\tau d_i \mid \Delta\tau d^j$. The weight column is a field that contains the value of the descriptor importance to facilitate the recognition. U_d is the thresholds vector used by the module ($\Gamma\rho_d$) to recognize its respective dynamic pattern. S_o (output signal): Each module produces a recognition signal (S_o), or petition signal toward lower levels.

Theorems of pattern matching for dynamic patterns

Now, we describe the theorems used in $\Gamma\rho_d$ to recognize a dynamic pattern. To recognize a dynamic pattern, its descriptors are evaluated in a time interval $[t_i, t_j]$. The general method of recognition involves verifying if the information of the

Table 1. Matrix $E_d = \;< S_d, C_d >$.

E_d			
S_d		C_d	
Signal	State	Pointer (P)[a]	Weight (W)
1	False	Pointer$_1$	[0,1]
…	…	…	[0,1]
N	False	Pointer$_i$	[0,1]
1	False	Pointer^1	
…	…	…	…
M	False	Pointer^j	[0,1]
		U_d: $< \Delta U1, \Delta U2 >$	

[a]An time series pointer in which we store the data.

input pattern is the same as the information stored in the time series. If the time series is consistent, it is reported as known (True). Otherwise, it is reported as unknown (False). The model works on thresholds based on time series

Theorem 1. Pattern matching strategy per key dynamic signals

This recognition uses the descriptors with greater weight of importance and the $\Delta U1$ threshold for the recognition.

Definition 1. A $S_i = (d_i \text{ or } d^j)$ dynamic signal is key if its importance weight has a value greater or equal to the average of all the signals in $\Gamma \rho_d$. This strategy uses the $\Delta U1$ threshold for the recognition.

$$\forall d_i \in \; S_{\Gamma_{\rho_d}}, \text{IF}\left[W(d_i) \geq W_{\text{average}}\left(S_{\Gamma_{\rho_d}}\right)\right] \rightarrow \text{Key}_{\Gamma_{\rho_d}}(d_i) \tag{8}$$

$$\forall d^j \in \; S_{\Gamma_{\rho_d}}, \text{IF}\left[W\left(d^j\right) \geq W_{\text{average}}\left(S_{\Gamma_{\rho_d}}\right)\right] \rightarrow \text{Key}_{\Gamma_{\rho_d}}\left(d^j\right) \tag{9}$$

Definition 2. A dynamic pattern is recognized by key dynamic signals of *characteristics* if:

$$\frac{\sum_{r=t_k}^{t_j} \sum_{i=1 \cap \text{ state}(d_i=\text{true}) \cap \; d_i \in \text{Key}_{\Gamma_{\rho_d}}}^{n} W(d_{ir})}{\left|\text{Key}_{\Gamma_{\rho_d}}\right|} \geq \Delta U1 S_o \tag{10}$$

If the weights reach the $\Delta U1$ threshold of recognition, then an output of pattern recognition is generated that changes its state to "True" in those higher-level patterns that contains them.

Definition 3. A dynamic pattern is recognized by key dynamic signals of *perception* if:

$$\frac{\sum_{r=t_j}^{t_k} \sum_{i=1 \cap \text{ state}(d^i=\text{true}) \cap \; d^i \in \text{Key}_{\Gamma_{\rho_d}}}^{n} W(d_r{}^j)}{\left|\text{Key}_{\Gamma_{\rho_d}}\right|} \geq \Delta U1 S_o \tag{11}$$

Theorem 2. Pattern matching strategy per partial mapping

In the case of pattern matching for partial mapping, this strategy consists of validating that a signals number present in $\Gamma \rho_d$ is superior to the $\Delta U2$ threshold.

Definition 4. A dynamic pattern is recognized by partial mapping of characteristic if:

$$\frac{\sum_{r=t_k}^{t_j} \sum_{i=1 \cap state(d_i=\text{true})}^{n} W(d_{ir})}{n} \geq \Delta U2S_o \tag{12}$$

If the weights reach the $\Delta U2$ threshold of recognition, then an output of pattern recognition is generated that changes its status to True in those higher-level patterns that contains them.

Definition 5. A dynamic pattern is recognized by partial mapping of perception if:

$$\frac{\sum_{r=t_j}^{t_k} \sum_{i=1 \cap state(d^i=\text{true})}^{n} W(d_{ir})}{n} \geq \Delta U2S_o \tag{13}$$

The algorithm of the model for recognition of patterns inspired in the brain, has seen presented in Puerto and Aguilar (2016b). AR2P exploits the idea of recursivity in the recognition process, and unbundling/integration of the pattern to recognize. Two analytical processes characterize the algorithm: a first process, called Top-Down, for recognition of the input pattern by decomposition: The top-level module invokes the modules of recognition of lower-level constituent, and these recursively do the same. A second process, called Bottom-Up, is for the recognition of atomic patterns: the output signal of the recognized pattern goes to the modules, of which it is part of top-level. Those top levels will be activated or not, if they pass the recognition threshold described.

Case studies

Recognition of "galloping horse" by dynamic patterns oriented to perception (DpoP)

The task is to recognize the pattern shown in Figure 1. It is assumed that the pattern consists of a sequence of discrete images recorded in 8 sec, one per second.

In this case study, the recognition by perception is analyzed. At the level of perception, we have the following *DpoP*: the tail, the legs, the body, and the head. The dynamic of this pattern (horse) is guided by changes in function of the horse's dynamic stance defined by the descriptors mentioned. Suppose the Eq. (1), $\rho_{d=horse} = < D_n, f_\Delta, \Delta\tau>$.

Using Eq. (2), the vector D_n with the collection of all the n descriptors *DpoP* of $\rho_{d=horse}$ is:

$$D_n = \left[d^{tail}, d^{legLeftD}, d^{legRightD}, d^{legLeftT}, d^{legRightT}, d^{Body}, d^{Head} \right]$$
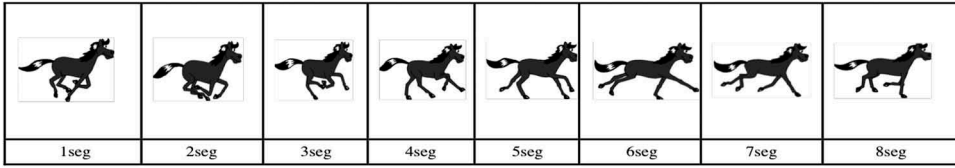
Figure 1. Galloping horse.

Now, the Eq. (4) is $\widehat{d^y} \equiv [v^{1y}, v^{2y}, v^{3y}, \ldots, v^{ky}]$. Remember $\widehat{d^y}$ is the domain vector of d^y (y is the perception descriptor) and has a range of values from v^{1y} to v^{ky}, with y = 1,2,3... to an arbitrary value, depending on the domain. Each value represents a dynamic value. For example, for the descriptor of the tail, when $v^{11} = 1$ will mean tail up, $v^{12} = 2$ means tail straight, and $v^{13} = 3$ means tail below. That is similar for the other descriptors.

$$\widehat{d^{tail}} \equiv [1, 2, 3]$$

$$\widehat{d^{legRightD}} \equiv [1, 2, 3, 4, 5, 6, 7, 8]$$

$$\widehat{d^{legLeftD}} \equiv [1, 2, 3, 4, 5, 6, 7, 8]$$

$$\widehat{d^{legRightT}} \equiv [1, 2, 3, 4, 5, 6, 7, 8]$$

$$\widehat{d^{legLieftT}} \equiv [1, 2, 3, 4, 5, 6, 7, 8]$$

$$\widehat{d^{Body}} \equiv [1, 2]$$

$$\widehat{d^{Head}} \equiv [1, 2, 3]$$

The eight values v_{ij} in the leg in the descriptors represent the eight possible positions of the legs, which can be described in different ways, for example, by internal angles, its distance to the body, its shape; for example, $v^{11} = 1 =$ leg stretched, $v^{12} = 2 =$ leg collected, etc.

Additionally, the change functions are defined. These functions model the transition (possible change) of the values of each descriptor (according to theirs domains). We define a change function for each descriptor.

$$\Delta = [fcd^{1tail}, fcd^{2legRightD} \ldots fcd^{7Head}]$$

Finally, each one of the descriptors may have a different time rate of change.

$$\Delta \tau d^{1tail} = [(t_1 = 0, 5s, v^{11} = 1s), (t_2 = 1s, v^{21} = 2), (t_3 = 1, 05s, v^{31} = 3)$$

$$\Delta \tau d^{1legRightD} = [(t_1 = 0, 3s, v^{11} = 1), (t_2 = 0.4s, v^{21} = 2), (t_3 = 0, 5s, v^{31} = 3),$$
$$(t_4 = 0, 6s, v^{31} = 4), (t_5 = 0, 7s, v^{31} = 5), (t_6 = 0, 8s, v^{31} = 6),$$
$$(t_7 = 0, 9s, v^{31} = 7), (t_8 = 10s, v^{31} = 9)]$$

$$\Delta\tau d^{1legLeftD} = \left[(t_1 = 0,2s, v^{11} = 2), (t_2 = 0.4s, v^{21} = 3), (t_3 = 0,5s, v^{31} = 4),\right.$$
$$(t_4 = 0,6s, v^{31} = 5), (t_5 = 0,7s, v^{31} = 6), (t_6 = 0,8s, v^{31} = 7),$$
$$\left.(t_7 = 0,9s, v^{31} = 8), (t_8 = 10s, v^{31} = 1)\right]$$

$$\Delta\tau d^{1legRightT} = \left[(t_1 = 0,1s, v^{11} = 2), (t_2 = 0.4s, v^{21} = 3), (t_3 = 0,5s, v^{31} = 1),\right.$$
$$(t_4 = 0,6s, v^{31} = 5), (t_5 = 0,7s, v^{31} = 6), (t_6 = 0,8s, v^{31} = 7),$$
$$\left.(t_7 = 0,9s, v^{31} = 8), (t_8 = 10s, v^{31} = 4)\right]$$

$$\Delta\tau d^{1legRightT} = \left[(t_1 = 0,3s, v^{11} = 2), (t_2 = 0.4s, v^{21} = 3), (t_3 = 0,5s, v^{31} = 4),\right.$$
$$(t_4 = 0,6s, v^{31} = 5), (t_5 = 0,7s, v^{31} = 6), (t_6 = 0,8s, v^{31} = 7),$$
$$\left.(t_7 = 0,9s, v^{31} = 8), (t_8 = 10s, v^{31} = 1)\right]$$

$$\Delta\tau d^{1body} = \left[(t_1 = 0,3s, v^{11} = 2), (t_2 = 0.4s, v^{21} = 1),\right]$$

$$\Delta\tau d^{1Head} = \left[(t_1 = 0,4s, v^{11} = 2), (t_2 = 0.6s, v^{21} = 1), (t_2 = 0.7s, v^{21} = 3)\right]$$

Remember, we want to recognize the galloping horse of the Figure 1. The recognition should be given in any of the eight times, which would result in "I'm seeing a horse gallop".

To continue, we analyze the last two top levels of the hierarchy (see Figure 2): at the level X_a we have Γ_{ja} modules for the pattern recognition of parts of an animal (*LegRightD,LegLftD, LegRightT, LegLeftT, Tail, Body, Head*), and level X_u, modules for the pattern recognition of an animal (such as *horse, cow*).

Now, we execute our general recognition algorithm (see AR2P algorithm in Puerto and Aguilar (2016b)). The process receives as input $y = s() = $ "the image of the horse in the first second". A decomposition of the image is then performed in its subpatterns < *LegRightD,LegLftD, LegRightT, LegLeftT, Tail,*
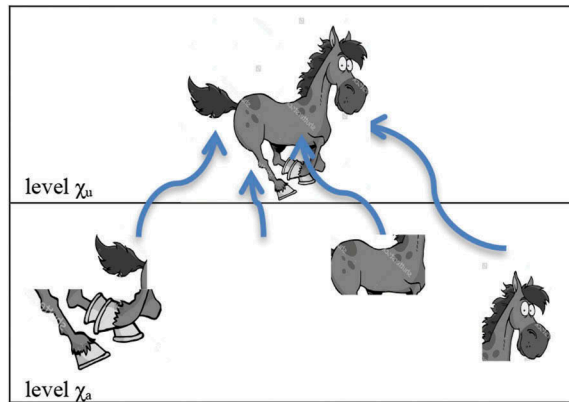

Figure 2. Three levels of the hierarchy of patterns, to recognize the pattern "horse".

Body, Head>. Then, it is determined the X_i level of the hierarchy where the process begins the recognition of (*y*), in this case $X_{i=2}$ (level of the animals).

At this level ($X_i = X_u$), their modules are instantiated for each instant of time. For example, the Table 2 shows an instance of the data structure of the image "horse" corresponding to the first image (first second). Then, L requests of petition for recognition of y are created, <*y1 = legRightD, y2 = LegLeftD, y3 = LeftRightT, y4 = LeftLeftT, y5 = body, y6 = tail, y7 = Head>*, in the next level.

Assuming the patterns (*tail, legleftD, lefRightD, . . .*) were recognized at the level X_a, they change the states of the superior levels of the hierarchy (level X_u). This level receives these recognized signals, and calculates the recognition using the theorems proposed:

Let $\Delta U1 = 0.85$, and that all signals were recognized (their status are "True", see Table 2). Then, we can recognize by perception (DpoP) according to the Eqs. (11)-(13). According to Eq. (11), the key signals of this pattern are: signals 1, 6, and 7. Now, we use Eq. (13) to determine if the pattern can be recognized with these recognized signals. These signals are sufficient to overcome the threshold (0.86 > 0.85). Pattern recognition is successful, and because is the last level of the hierarchy X_u (the level where began the process of recognition), it produces the output signal S_o, and generates the system output signal: "S0 = I'm seeing a horse gallop".

Recognition of the emotions "anger" or "happiness", by dynamic patterns oriented to characteristics (DpoC)

In this case, we recognize the emotional patterns of a car driver using dynamic patterns oriented to characteristics (DpoC). We define three physiological descriptors with the value of the Table 3 for each emotion, in order to test the emotion recognition in our system: one for when the car driver is happy, and another for when the car driver is with anger.

Table 2. Matrix E_d = < Galloping horse>.

E_d				
S_d			C_d	
Signal	State		Pointer (P)[a]	Weight (W)
$1 = d^{tail}$	T		Pointer[1]	0.9
$2 = d^{LegRightD}$	T		Pointer[2]	0.6
$3 = d^{LegLeftD}$	T		Pointer[3]	0.6
$4 = {}^{LegRightT}$	T		Pointer[4]	0.6
$5 = d^{LegRightT}$	T		Pointer[5]	0.6
$6 = d^{body}$	T		Pointer[6]	0.8
$7 = d^{Head}$	T		Pointer[7]	0.9
		U_d: < 0.85, 0.65 >		

Table 3. Physiological conditions for the emotional patterns of a car driver.

Physiological pattern of a car driver	Happy	Heart rate	From [0.1 to 0.4]
		Respiratory rate	From [0.1 to 0.4]
		Blood pressure	From [0.1 to 0.4]
	Anger	Heart rate	From [0.5 to 1.0]
		Respiratory rate	From [0.5 to 1.0]
		Blood pressure	From [0.5 to 1.0]

All values are normalized in [0, 1]. It is assumed that the pattern consists of a sequence of values (heart rate, respiratory rate, or blood pressure) of the emotional state of the car driver in a time interval Δt. Suppose the Eq. (1), $\Gamma_{\rho d= \text{joy}} = < D_n, f_\Delta, \Delta\tau>$. Using Eq. (2), the vector D with the collection of all the n descriptors DpoC of $\Gamma_{\rho d=\text{happy}}$ is:

$D_{n(\text{happy})} = [d_{\text{hearate}}, d_{\text{respiratorate}}, d_{\text{bloodpressure}}]$ In the same way for $\Gamma_{\rho d=\text{angry}}$

$$D_{n(\text{angry})} = [d_{\text{hearate}}, d_{\text{respiratorate}}, d_{\text{bloodpressure}}]$$

Tables 4 and 5 show the structure of the recognition module for each emotion

To recognize a dynamic pattern, its descriptors are evaluated in a time interval $[t_{i=1}, t_{j=n}]$. Suppose the next time series for Anger (see Eq. (6)).

$$\Delta t_{\text{hearate}}(\text{anger}) = [(t1, 0.5), (t2, 0.6), (t3, 0.7), (t4, 0.8)]$$

$$\Delta t_{\text{respitatorate}}(\text{anger}) = [(t1, 0.5), (t2, 0.6), (t3, 0.7), (t4, 0.8)]$$

$$\Delta t_{\text{bloodpressure}}(\text{anger}) = [(t1, 0.5), (t2, 0.6), (t3, 0.7), (t4, 0.8)]$$

And for happy.

$$\Delta t_{\text{hearate}}(\text{happy}) = [(t1, 0.5), (t2, 0.4), (t3, 0.3)]$$

$$\Delta t_{\text{respitatorate}}(\text{happy}) = [(t1, 0.5), (t2, 0.4), (t3, 0.3)]$$

$$\Delta t_{\text{respitatorate}}(\text{happy}) = [(t1, 0.5), (t2, 0.4), (t3, 0.3)]$$

Suppose the next input $s() = \{$hearate$[(t1, 0.5), (t2, 0.6), (t3, 0.7), (t4, 0.8)]$, respiratorate$[(t1, 0.5), (t2, 0.5), (t3, 0.6), (t4, 0.8)]$, bloodpressure$[(t1, 0.5), (t2, 0.6), (t3, 0.7), (t4, 0.7)]\}$ from a car driver.

Each element of this set is an atomic input pattern. Each pattern is recognized individually, through a direct mapping to the existing set of

Table 4. Matrix $\Gamma\rho_{d = \text{angry}}$.

E_d			
S_d		C_d	
Signal	State	Pointer (P)	Weight (W)
1	F	PointerAngry$_{1\text{hearate}}$	1
2	F	PointerAngry$_{2\text{respitarotate}}$	0.8
3	F	PointerAngry$_{3\text{bloodpressure}}$	0.9
		U_d: $<\Delta U1, \Delta U2 >$	

Table 5. Matrix $\Gamma\rho_{d\,=\,happy}$.

S_d		C_d	
Signal	State	Pointer (P)	Weight (W)
1	F	PointerHappy$_{1hearate}$	0.8
2	F	PointerHappy$_{2respitatorate}$	0.9
3	F	PointerHappy$_{3bloodpressure}$	0.8
		U_d: $<\Delta U1, \Delta U2>$	

time series (via the pointer). Of the three patterns, only *hearate* is recognized, which match the pattern of heart rate of the anger emotion. This causes a change in the first signal of the recognition module of anger, of false to true (see Table 6).

Let $\Delta U1 = 0.9$. Then, we can recognize by characteristic (DpoC) according to the Eqs. (10–12). According to Eq. (10), the key signals of this pattern are the first and third signals. Now, we use Eq. (12) to determine if the pattern can be recognized with the recognized signals (first signal). These signals are sufficient to overcome the threshold ($1 > 0.9$). Pattern recognition is successful, and how it is the last level of the hierarchy X_u (the level where began the process of recognition), it produces the output signal S_o, that becomes in **the system output signal**: "$S_o = angry$".

Conclusions

This paper presents a formal model for the dynamic and static pattern recognition, based on the theory of mind proposed in (Kurzweil 2012). The proposed model is a computational approach to pattern recognition based on the functional mechanisms of the human brain.

Our pattern recognition model is highly recursive and uniform. The uniformity allows the recursive process to be the same in all the levels. Thus, the model recognizes a pattern of input no matter their level of complexity or the nature of the same (static or dynamic). The model recognizes the input patterns by a process of self-association in a pattern hierarchy.

Table 6. Matrix $\Gamma\rho_{d\,=\,angry}$.

S_d		C_d	
Signal	State	Pointer (P)	Weight (W)
1	**F**	**PointerHappy$_{1hearate}$**	**1**
2	F	PointerHappy$_{2respitatorate}$	0.9
3	F	PointerHappy$_{3bloodpressure}$	0.8
		U_d: $<\Delta U1, \Delta U2>$	

The model can recognize a dynamic pattern from two major orientations: recognition of dynamic patterns oriented by characteristics, or by perceptions. In addition, the model uses two strategies for recognition; one for key signals and another for total or partial mapping. The first exploits the important signals that facilitate the recognition; the second strategy completely exploits the input signals. In the key signal strategy, it can predict a pattern even with missing information.

Another important feature of the model that makes it different from other recognition systems is its ability to recognize a pattern using the natural dynamic of the pattern. Specifically, it uses time series and automata for that.

At the computational level, the proposed model offers a concise, readable and elegant solution (recursivity), but it defines a tree, which can be very large. One possible optimization is to use parallelism to improve the execution time. Additionally, a main problem is to define the atomic level for each pattern. Future works will be dedicated to extend this model considering both aspects has not been published elsewhere and that it has not been submitted simultaneously for publication elsewhere. Finally, we state that this manuscript has not been published elsewhere and that it has not been submitted simultaneously for publication elsewhere.

Acknowledgments

ORCID

Eduard Puerto http://orcid.org/0000-0001-9361-5837
Jose Aguilar http://orcid.org/0000-0003-4194-6882
Danilo Chávez http://orcid.org/0000-0002-7529-9006

References

Aguilar, J. 2004. A color pattern recognition problem based on the multiple classes random neural network model. *Neurocomputing* 61:71–83.

Aguilar, J., and A. Colmenares. 1997. Recognition algorithm using evolutionary learning on the random neural networks. In IEEE *International Conference on Neural Networks*, vol. 2, pp. 1023–28. Houston, USA.

Alpaydin, E. 2014. *Introduction to machine learning*. Cambridge, England: MIT press.

Bobrow, J. 2014. *Representation and understanding: Studies in cognitive science*. New York, USA: Elsevier.

Felzenszwalb, P., D. McAllester, and D. Ramanan. 2008, June. A discriminatively trained, multiscale, deformable part model. *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* (pp. 1–8). Anchorage, USA: IEEE.

Hawkins, J., and S. Blakeslee. 2007. *On intelligence*. New York, USA: Macmillan.

Kaku, M. 2014. *El futuro de nuestra mente*. Barcelona: Debate.

Kelso, J. S. 2014. The dynamic brain in action: Coordinative structures, criticality and coordination dynamics. In *Criticality in Neural Systems*, ed. D. Plenz and E. Niebur, 67–104. Berlin, Germany: Wiley.

Kurzweil, R. 2012. *How to create a mind: The secret of human thought revealed*. New York, USA: Penguin.

LeCun, Y., Y. Bengio, and G. Hinton. 2015. Deep learning. *Nature* 521 (7553):436–44.

Liu, C., B. Lovell, D. Tao, and M. Tistarelli. 2016. Pattern recognition, part 1. in: IEEE Intelligence Systems. *IEEE* 31 (2):6–8.

Lopes, N., and Ribeiro, B. 2015. Machine Learning for Adaptive Many-Core Machines-A Practical Approach. New York, USA: Springer International Publishing.

Markram, H. 2012. The human brain project. *Scientific American* 306 (6):50–55.

Moser, M. B., and E. I. Moser. 2014. Understanding the Cortex through Grid Cells. In *The future of the brain: Essays by the world's leading neuroscientists*. ed. G. Marcus, and J. Freeman, 67–77. New Jersy, USA: Princeton University Press.

National Institutes of Health. The brain initiative, US. Accessed October 8, 2016. hhttp://www.braininitiative.nih.gov/2025/index.htm.

Pavlidis, T. 2013. *Structural pattern recognition*, Vol. 1. Berlin, Germany: Springer.

Puerto Cuadros, E. G., and J. L. Aguilar Castro. 2016a. Learning algorithm for the recursive pattern recognition model. *Applied Artificial Intelligence* 30 (7):662–78.

Puerto, E., and J. Aguilar. 2016b. Formal description of a pattern for a recursive process of recognition. *Computational Intelligence (LA-CCI), 2016 IEEE Latin American Conference on*. pp. 1–2. Cartagena, Colombia.

Srivastava, N., R. R. Salakhutdinov, and G. E. Hinton. 2013. Modeling documents with deep boltzmann machines, *Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, pp. 616–624. Arlington, USA.

Taubman, D., and M. Marcellin. 2012. *JPEG2000 image compression fundamentals, standards and practice*. New York, USA: Springer.

Watanabe, S., Ed. 2014. *Methodologies of pattern recognition*. New York, USA: Academic Press.

Xu, T., Z. Yang, L. Jiang, X. X. Xing, and X. N. Zuo. 2015. A connectome computation system for discovery science of brain. *Science Bulletin* 60 (1):86–95.