

# Minimal and necessary conditions for the emergence of species-specific recognition patterns

Nicolas Brodu<sup>1,2</sup>

<sup>1</sup> INRIA-IRISA, Campus de Beaulieu, 35042 Rennes Cedex, France

<sup>2</sup> Concordia University, 1455 de Maisonneuve Blvd. W. Montreal, QC H3G 1M8, Canada

nicolas.brodu@free.fr

**Abstract.** A simple mechanism is presented for the emergence of recognition patterns that are used by individuals to find each other and mate. The genetic component determines the brain of an individual, a machine learning architecture which is then used to transmit knowledge. Thanks to the interactions between the genetic and the knowledge parts the agents get to use species-specific recognition patterns, starting from an initial condition where the species are not distinguishable. Several machine learning architectures are investigated, as well as the influence of space and asynchronous genetic algorithm operations. Agents selecting each other for mating based on their limited recognition capacities is all that is needed for the emergence of species-specific recognition patterns: the transition between symbols to sequences with an intrinsic role within the species.

**Keywords:** Mutual recognition, Knowledge transmission, Genetic algorithm, Baldwin effect, Communication.

## 1 Introduction

Computer simulations are powerful tools to analyze the emergence of language [1], but despite the progress they entail [2] the field remains controversial [3, 4, 5, 1]. The work introduced here is about interactions between communication and reproduction. Previous related work have studied for example the knowledge transmission of the categorization of object attributes [6], or have introduced specific mappings between meaning and symbols [7]. The present work does not rely on any a priori semantic concepts. The model is stripped down to the bare minimum: genetic reproduction, and basic learning capabilities. The model also omit social interactions [8, 9] and ecology [10]. Yet complex patterns emerge for the mutual recognition of individuals belonging to the same species. The goal is then to find the necessary and sufficient conditions for the emergence of these mutual recognition patterns. These patterns may perhaps serve as basis for a protolanguage [11,5], which may then be extended into a full-featured language thanks to social interactions [9]. This work is about how some of the precursor patterns may form in the first place, the transition from isolated symbols to sequences, not about the later two transitions to a full-featured language.

Communication is imperfect and takes the form of strings of symbolic values. Each individual emits a string, and it is presented with the strings from the other

individuals. The task is then to find a suitable mate, which necessarily implies the formation of specific patterns for higher than random recognition rates. By design less symbols are available than the number of species, so mono-symbol sequences cannot uniquely identify a species and more complex patterns are required. As time passes the agents get to recognize each other better, using these more elaborated strings.

Genetics alone cannot determine the recognition patterns for the simplest models. Knowledge transmission alone cannot solve the problem either: there are few and noisy learning instances. Hence the combination of both is necessary for the agents to agree on more complex recognition patterns. Species are initially indistinguishable. The individuals who could find a mate may teach the others, according to the algorithm presented in Section 2 and 3. Genetics act on the brain structure. Several simple machine learning models are compared so as to determine the minimal conditions for the emergence of the recognition patterns. These models are detailed Section 4. Results are presented in Section 5, and the role of space is then investigated in Section 6, together with the influence of synchronizing or not the genetic algorithm operations in time. Section 7 concludes this work and proposes possible extensions.

## 2 The model

The goal of this experiment is to investigate the minimal and necessary conditions for the emergence of species recognition patterns. A simulation model is built accordingly: Each agent is equated to an AI model for performing string recognition and symbol production. The parameters of this AI model form the genome for the agent (See Fig. 1). Several AI models are used, they are presented in Section 3.

Each agent produces a “song”. Each agent then selects a mate according to how much it likes the other agents songs. Only agents choosing a mate from the same species may reproduce. The symbols that are used to build the songs are assumed to be available and identical for all agents. Agents communicate only through the sequence exchange. In particular, there is no way for an agent to assert the species of another agent except by inferring this information from the symbol sequence of the other agent. So, this model operates on the transition from isolated symbols to recognition patterns ; it aims at providing some reasons why a structure may appear in

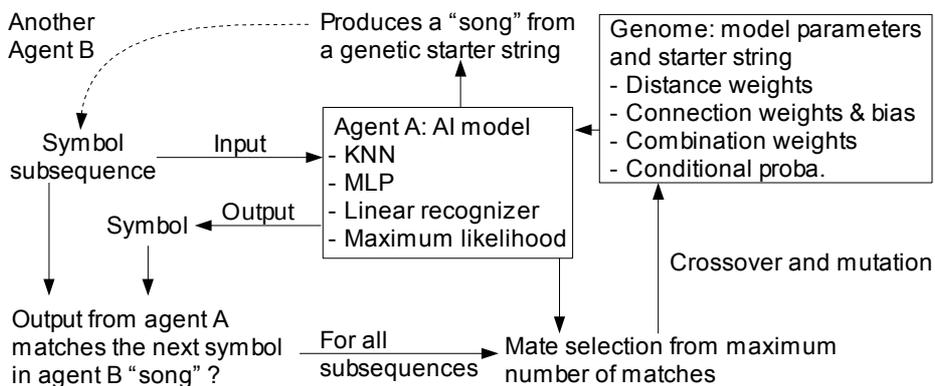


Fig. 1. The simulation model.

the symbol sequences. In a first time, the model is simplistic with discrete time (synchronous genetic algorithm) and no spatial structure. Section 6 then extends the model to continuous time (asynchronous genetic algorithm) and three-dimensions, in order to alleviate the fairly strong assumptions of the discrete model.

### 3 The genetic algorithm / machine learning interactions

The core of the algorithm is a feedback loop between the genetic and the machine learning algorithm components: Each individual must recognize a mate for the selection process, amongst all individuals from all species, and only successful individuals may reproduce. There is thus no explicit fitness function. Each individual produces a “song” that will be presented to the others during a “mating parade” (see Fig. 1). The discrete time, synchronous genetic algorithm is sketched in Fig. 2.

```

1 for each generation
2   All individuals produce a "song"
3   for each individual
4     listen to all other songs, choose one mate
5     if same species => put in reproducers set
6   for maximum R reproducers at random per species
7     mate to produce a child
8     child trains on both parent songs
9     selected partner trains on reproducer song
10    replace one individual at random by the new child

```

**Fig. 2.** The core algorithm where both genetic and knowledge components interact. The main text introduces the algorithm details, like R which is the maximum turnover rate.

More formally, a song is a series of symbols,  $(s)_i$  with  $i=1..M$  and  $M$  the maximum song size. Random series are presented to the first generation in order to bootstrap the experiment. A moving window of size  $N$  inputs is applied to each substring  $(s)_j$  of each training song, with  $j=i..i+N-1$ . The individual is trained to produce  $s_{i+N}$  the next symbol of the sequence for each such substring (see Fig. 3). For the first symbols  $i = -N..-1$  of the sequence there are not enough previous symbols to fill the moving window and genetically determined symbols fill the substring (Fig. 3). The generation of the songs is the natural reverse operation: The genetic starter string is presented and the individual is asked to produce a symbol based on its previous knowledge.

Some genetic starter strings  $G$  are better suited than others for learning some Song Production

- Initialize with genetic starter  
 - Mappings from moving window  
 - Produce up to N symbols

String→Symbol mapping by AI

$+XXO+ \rightarrow +$	$++XO+ \rightarrow X$	$+XXO+ \rightarrow X$	...	Agent 1 song
$XXO++ \rightarrow X$	$+XO+X \rightarrow +$	$+XXO++ \rightarrow X$	$\rightarrow +XO+XO+ \dots$	
$XO++X \rightarrow O$	$XO+X+ \rightarrow O$	$OXX+X \rightarrow O$	...	Agent 2 song
$O++XO \rightarrow +$	$O+X+O \rightarrow +$	$OXX+XO \rightarrow O$	$\rightarrow OX+OO+OX \dots$	

#### Mate Selection

Agent 1 notes Agent 2 song by checking what it would have produced with the same strings. Agent 1 selects the mate with most points.

Starter 1	Song 2	Agent 2: 5 points
$+XXO+OX+OO+OX+$	$+XXO+OX+OO+OX+$	$+O+XO+XX+$
		$+O+XO+XX+$
		$O+XO+XX+$
		$O+XO+XX+$
		$O+XO+XX+$

**Fig. 3.** Song production and mate selection from symbol sequences.

sequences. Suppose  $G=AAAA$  and the task is to learn the song  $AB$ . In this case, the system will generate two conflicting training instances  $AAAA \rightarrow A$  and  $AAAA \rightarrow B$ . On the other hand if  $G=CDDC$  there is no conflict. The genetic part thus has an influence on how well it is possible to learn some sequences. Conversely there are several possible genetic starters which are all equally suited with respect to learning a particular song. Hence the genome does not determine the species song, it merely defines for each individual a subspace of all possible sequences for which there is no conflict. Any song within these subspaces intersection can be learned equally well by all members of the species, even if they have different genetic material.

In the mate selection task each agent classifies the candidate songs by order of preference. Each candidate song is fed as input, possibly after alteration by imperfect communication (symbols are modified at random with a predetermined probability). The individual then estimates what symbol it would have produced for each substring (see Fig. 3). When the symbols match the candidate gets one point. Each individual selects a mate with maximal points, choosing at random for ex-aequo candidates. The agent may reproduce if the selected partner belongs to the same species.

The mating process is straightforward: crossover and mutation of the AI model parameters between the parents so as to produce the offspring. But at this point the new child has no training and is thus unable to produce its own sequence at the next generation. The parents songs are used so a child starts with only two training instances (possibility imperfectly communicated). But now, if all individuals from one generation successfully reproduce they are all killed and replaced by their offspring, destroying the knowledge that accumulated with time. In order to eliminate this risk a maximum number of children  $R$  is introduced in each species. With this setup the individuals get a chance to survive more than one generation and accumulate knowledge. However no mechanism has yet been introduced that would allow this accumulation (only one that prevents the non-accumulation). In order to get more training instances a selected mate also trains on the reproducer which selected it (Fig. 2). Thus, as time passes, agents now have a chance to accumulate knowledge.

## 4 The different machine learning models

Any machine learning model may be used in the algorithm in Fig. 2. This model receives as training instances (sequence, symbol) pairs, and must predict a symbol when presented other (possibly unknown) sequences. The models chosen for this study are simple ones, exploiting different information, since the goal is to investigate what are the minimal conditions for the emergence of mutual recognition patterns.

### 4.1 The linear classifier

As often for categorical data (the symbols) each input is duplicated into  $L$  entries, with  $L$  the size of the alphabet. Each of these entries is here set to  $+1$  or  $-1$  if the corresponding input matches that entry value. For example: with  $L=3$  symbols  $ABC$ , the input string  $ABCA$  is mapped into the 12 entries vector  $I=(+1,-1,-1, -1,+1,-1, -1,-1,+1, +1,-1,-1)$ . Similarly the vector  $P$  for the symbol to predict contains  $S$  outputs.

The training instances are converted in this (I,P) format. For T training instances the I vectors form a  $N.L \times T$  matrix A and the P vectors a  $L \times T$  matrix B. The least squared error solution for the equation  $WA=B$  gives the weights W that are used by the linear classifier. Then, for a new unknown instance J, the predicted vector is  $P=WJ$ . The output symbol for J is extracted from P as the one with largest entry. For example suppose  $P=(0.68, -0.87, 0.05)$ , then the symbol A is returned. When two symbols have equal value one is chosen at random, which makes the machine learning algorithm occasionally non-deterministic. This is acceptable in our context, especially since any song may be altered randomly later on by imperfect communication anyway. The linear classifier model has no genetic component in addition to the initial sequence of symbols presented in the previous section.

#### **4.2 The 2-layer perceptron (MLP)**

The same setup as for the linear classifier is reused for mapping the symbols to categorical data. A 2-layer perceptron then processes the input data. More precisely, the N.L categorical entries are connected to the input neurons. This input layer is connected to 10 hidden neurons with sigmoidal transfer function ( $f(x) = x / (1 + \text{abs}(x))$ ) is used here for its reduced computational costs compared to the more usual tanh, see [12]). An output layer with linear activation functions finally maps the results of the hidden layer to the L output categorical entries. The training set is formed as before. The MLP is trained simply by performing 30 steps of batch gradient descent with a learning rate of 0.1 over all known instances.

The MLP initial connection weights and biases before learning form an additional genetic component, together with the initial sequence of symbols presented in the previous section. When the training set is fixed (i.e. when individuals have agreed on a unique species recognition pattern) then individuals who have a genetic information (initial weights) that is better suited to this training set have an advantage over the others since they need less training, hence a Baldwin effect is expected [13].

#### **4.3 The K-nearest neighbors(KNN) classifier, with K=5**

The KNN model represents a simple form of learning by imitation of previously observed instances. [14] notes that “simple models of cultural transmission solely based on imitation are not sufficient to permit linguistic co-ordination”. However, as mentioned in the introduction, the current work is not about the emergence of a full featured language, just about the emergence of recognition patterns. The more elaborated mechanisms that would additionally be necessary to turn these precursor patterns into a full language are out of scope. However the only way to assert whether the KNN model – imitating previous instances – is sufficient for the emergence of the recognition patterns, is to test it in practice.

Each training sequence  $S=(s)_i$  with  $i=1..N$  is kept with the associated next symbol X, forming a pair (S,X). When a symbol has to be predicted from an unknown input sequence  $\Sigma$ , the distance between  $\Sigma$  and each known S is computed. That distance is simply the number of differences between  $\Sigma$  and S. For example, ABCD and AACD are at distance 1, ABCD and BBCA are at distance 2, etc. The K nearest S are then

selected, with ex-aequo chosen at random if necessary.

Then, for each of the up to K neighbors, the symbol X associated to that neighbor S is given a weight  $v_k$ . This weight  $v_k$  depends on the neighbor distance order k. Summing over all neighbors, the output symbol with the largest total weight wins the selection: It is returned by the classifier as the result of predicting the sequence  $\Sigma$ .

In this model the votes  $(v_k)_{k=1..K}$  associated to each of the K neighbors (in distance order) are genetic parameters in addition to the initial sequence of symbols presented in Section 2.  $K=5$  has been chosen for this study, though with the voting mechanism it may happen that some of the  $v_k$  become null during the genetic evolution and thus reduce the effective value of K.

#### 4.4 The assembly of maximum likelihood (ML) estimators

For a sequence  $S=(s)_i$ , a ML learner seeks to maximize the probability of this sequence  $p(s_1...s_N|t)$  over all possible output symbols t. Unlike the more usual approach of maximizing  $p(t|s_1...s_N)$ , the probability of obtaining t given the observed sequence, the maximum likelihood approach discriminates between competing sequences. The probabilities are noted from the samples, but unfortunately estimating  $p(s_1...s_N|t)$  requires monitoring  $L^{N+1}$  combinations (one  $L^N$  for each t), with L the number of symbols. A simple yet limited solution is to consider that inputs are independent, simplifying  $p(s_1...s_N|t)$  into  $\prod_{i=1..N} p(s_i|t)$ , hence reducing the complexity to  $N \times K^2$  combinations. An intermediary solution allowing one level of dependence has been chosen for this work. Inputs are gathered into mutually independent groups (assumption A1). A main input is chosen in each group, and the other group members are assumed to be independent conditionally to this input (assumption A2).

Example: Suppose  $N=5$ , with two groups  $\{s_1, s_2, s_3\}$  and  $\{s_4, s_5\}$ , and with  $s_1$  and  $s_4$  the group leaders. In this case:

$$p(s_1, s_2, s_3, s_4, s_5 | t) = p(s_1, s_2, s_3 | t) \cdot p(s_4, s_5 | t) \quad \text{Using A1}$$

$$p(s_1, s_2, s_3, s_4, s_5 | t) = p(s_2, s_3 | s_1, t) \cdot p(s_1 | t) \cdot p(s_5 | s_4, t) \cdot p(s_4 | t)$$

$$p(s_1, s_2, s_3, s_4, s_5 | t) = p(s_2 | s_1, t) \cdot p(s_3 | s_1, t) \cdot p(s_1 | t) \cdot p(s_5 | s_4, t) \cdot p(s_4 | t) \quad \text{With A2}$$

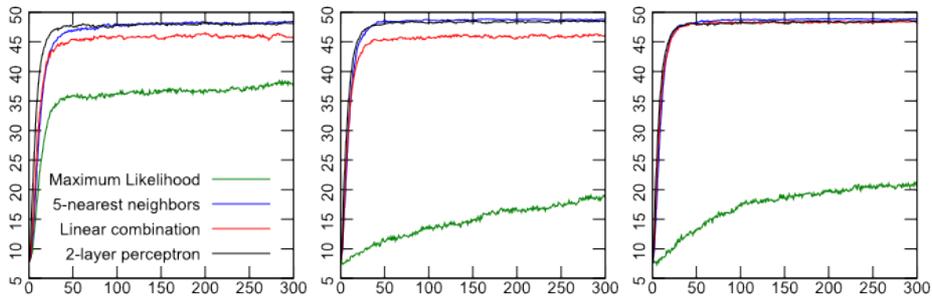
One level of dependence is thus kept, while maintaining the number of combinations to monitor in  $O(K^3)$  instead of  $K^{N+1}$ : one  $K^3$  for each  $p(s_j|s_i, t)$  with  $s_i$  a group leader and  $s_j$  in that group. An assembly of maximum likelihood predictors was introduced so as to deal with more complex songs: Several predictors are maintained in parallel, each with its own conditional dependence assumptions on the inputs. The final predicted symbol is simply the result of a majority vote amongst the predictors.

For each possible output symbol t,  $p(s_1...s_N|t)$  is computed using the decomposition presented above. The symbol with maximum likelihood value is selected. In the case where some  $p(s_i|t)$  were not observed, the selection operates between outputs with less unknown  $p(s_i|t)$ . This is equivalent to still noting known subsets of inputs when the whole sequence is unknown. When all input combinations are unknown no output is selected and majority is then voted amongst the other predictors in the assembly (which use different grouping assumptions). When all predictors fail the song simply stops. Ex-aequo situations are solved by choosing one candidate solution at random.

In this study 3 ML estimators are gathered in an assembly. The grouping information for the conditional dependence between the inputs form an additional genetic material together with the initial starter sequences.

## 5 Results for the synchronous case

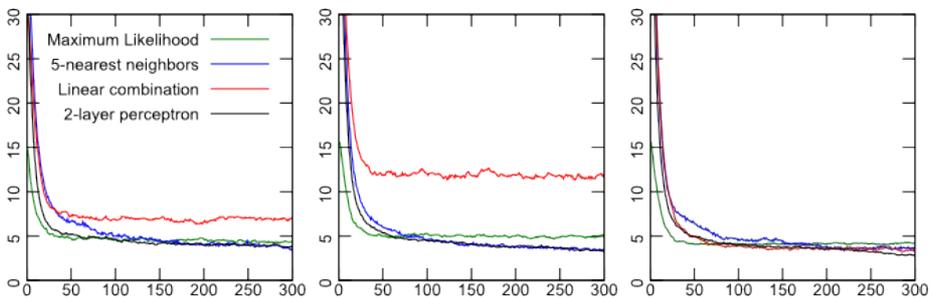
As all species receive equal treatment the results can be averaged over all species to give synthetic indicators for the whole population. The experiments in this section use: 6 species, 49 individuals per species, a moving window size of  $N=5$  inputs, a maximum song size of  $M=10$  symbols, and a maximum turnover rate of 20% (fixing  $R$  in Fig. 2). A first experiment is performed using 3 symbols. There is a probability of 0.01 that each time a symbol is transmitted it is replaced by another one at random. A second experiment reduces the number of symbols to 2, and a third experiment studies the effect of removing the transmission errors. 20 batches of runs are performed with the same random seeds for each experiment, and repeated again for all 4 machine learning algorithms. The results are plotted in Fig. 4-7.



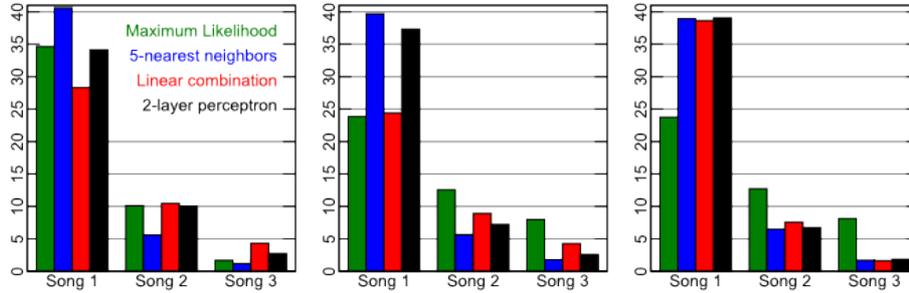
**Fig. 4.** Evolution of the number of reproducers vs. number of generations, when 2 symbols are in use (left), 3 symbols (middle), and 3 symbols with no transmission error (right).

Fig. 4 highlights the failure of the ML model to produce recognition patterns. The simplest linear classifier is less efficient than the KNN and MLP models in the noisy scenarios (left and middle plots). The number of symbols does not seem to influence much the models, except for the ML recognizer. An hypothesis would be a lack of training examples so to produce reliable statistics in the ML model, with more symbols meaning more combinations hence even less instances for each combination. Experiments performed where all individuals listen to all the species songs tend to confirm this hypothesis by improving the performance of the ML model.

Figure 5 introduces the number of songs used in each species. It is not obvious whether the present scenario converges or not to a unique song for each species, given the limited number of training instances for the children and the transmission errors.



**Fig. 5.** Evolution of the number of songs vs. the number of generations, when 2 symbols are in use (left), 3 symbols (middle), and 3 symbols with no transmission error (right).



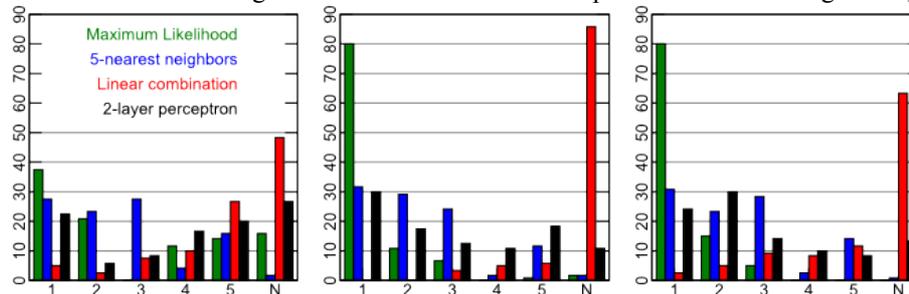
**Fig. 6.** Number of users for the three main songs in each species, when 2 symbols are in use (left), 3 symbols (middle), and 3 symbols with no transmission error (right).

Powerful AI models may also learn several songs. Once again the KNN and MLP models are relatively insensitive to both noise and number of symbols. The linear recognizer is too sensitive to noise, as is apparent from both Fig. 4 and Fig 5.

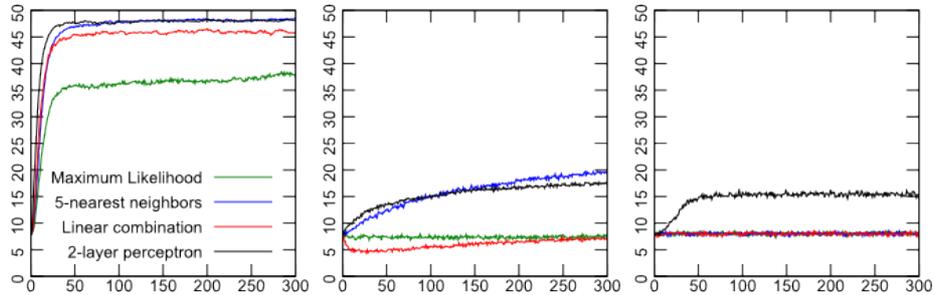
Figure 6 shows the repartition of the individuals using the few songs that are present in each species. In this synchronous scenario the dominant song is shared by a large majority of the individuals. The remaining songs are variants emitted probably by individuals without enough training (like the children). Some examples of dominant songs produced at the end of the 300 generations (with symbols noted as numbers) are the obvious mono-symbol sequences like 222222222, etc., the cycle-2 patterns like 0101010101, and other repetitive patterns like 0110110110, 1100011000, 1201201201, etc. The patterns may also be more complex, like 2221102212: Even though the fixed window size of  $N$  inputs would eventually make the trailing sequence in these patterns cyclic, the genetic starter string must be taken into account for determining the first symbols, which are thus not part of the eventual cycle, but nevertheless included in the pattern recognition between individuals.

Figure 7 displays the repartition of the songs according to their cycle length. The number of acyclic (over the first symbols) songs is highest for the linear recognizer, possibly due to the aforementioned sensitivity of that model. The ML model fails to produce distinctive patterns for each species, which corroborates Fig. 4: that model could not produce the more complex songs, necessary to overcome the symbol limit. Figure 5, right, shows that the MLP and the KNN have similar performances. Figure 7 shows however that the KNN classifier makes use of simpler recognition sequences on average, while the MLP produces a more diverse complexity repartition.

In order to investigate what are the intrinsic capabilities of each algorithm, a



**Fig. 7.** Percentage of songs with given cycle lengths in abscissa ( $N$  = no cycle), when 2 symbols are in use (left), 3 symbols (middle), and 3 symbols with no transmission error (right).



**Fig. 8.** Evolution of the number of reproducers vs. number of generations, with only knowledge transmission (middle) and with only the genetic algorithm (right). The left plot from Fig. 2 is reproduced with a similar scale to ease comparison.

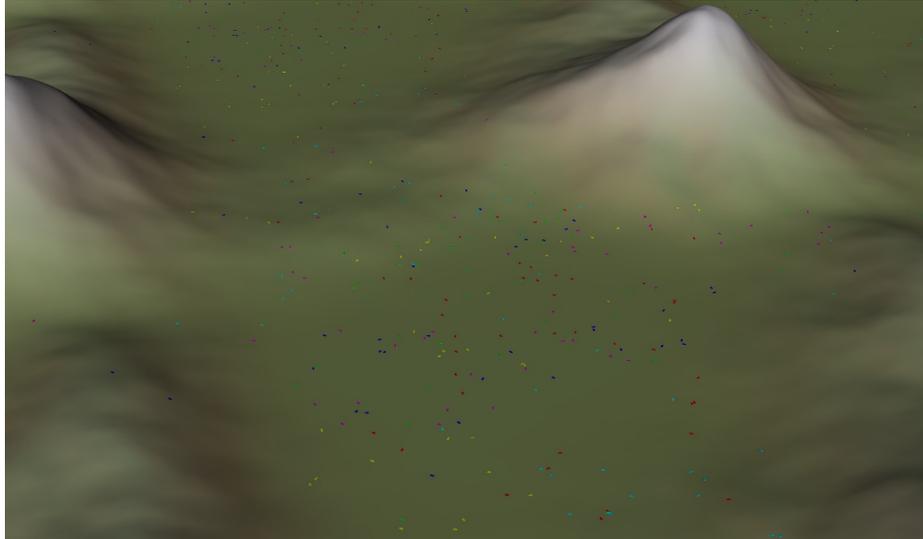
simple solution is to disable the genetic or the knowledge transmission part. Without knowledge transmission only the genetic structure may evolve, and without the genetic algorithm the initial agents may only learn from each other without producing new children. Figure 8 proves that both components are necessary for the emergence of efficient recognition patterns, though the two most successful models (KNN and MLP) still exhibit limited capabilities with only one component active. The interactions between the genetic and the knowledge transmission parts, however, are necessary for producing real recognition patterns: the levels obtained with the partial cases correspond to less than half the population successfully recognizing each other.

## 6 Results for the asynchronous case

The synchronous selection operation without spatial organization is useful for determining the respective influence of the models, but it imposes a fairly severe constraint on the genetic algorithm. Moreover these assumptions go against the goal of analyzing the minimal conditions for the emergence of the recognition patterns.

A more general framework is thus needed, where the influence of the spatial distribution of agents may be studied, together with the possibility for the agents to reproduce at any time. Figure 9 is a capture of the 3D simulation, with continuous space and time. The agents are embodied as vehicles with definite mass, position and velocity, and wander in the world with the aim of avoiding collisions. No further AI is given to the agents. Each agent chooses a mate as before, but only amongst neighbors present within a predetermined radius. The influence of space on the simulation is studied by varying the search radius. The agents reproduce at their own rhythm, determined by a frequency and a phase. Each agent has its own phase so the genetic algorithm operations are performed asynchronously in time.

Another change from the basic experiment is necessary due to the spatial localization: a minimum delay between reproduction events. This minimum delay ensures that a child has some time to move away from its parent, and that isolated mates don't reproduce too fast independently of the rest of the species. The asynchronous aspect is also enhanced, since the delays are randomly set for each reproduction event. A negative learning was finally introduced in the scenario, with agents learning instances that do not lead to a mating operation as bad sequences,



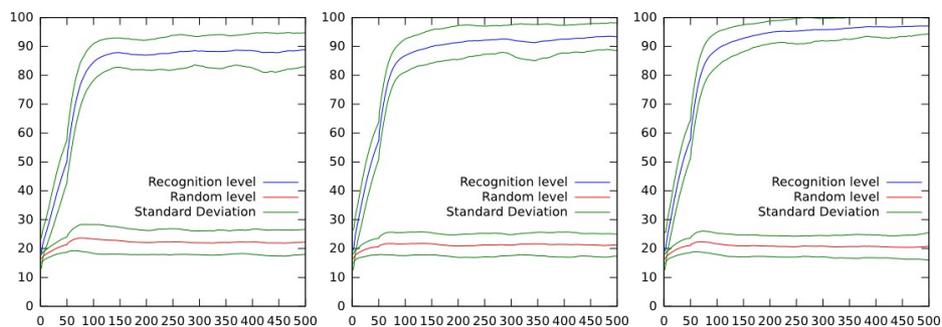
**Fig. 9.** Three-dimensional environment with an asynchronous genetic algorithm.

with the hypothesis that it could improve the species recognition.

Results for the recognition capabilities are given in Fig. 10, using the KNN learning algorithm, by averaging the results for 6 species over 20 experiments. These plots are the equivalent of Fig. 4 in the present asynchronous scenario.

The base random level is computed by checking how much agents from the same species were present in each neighborhood at each reproduction event; it gives the chance an agent would select another one from the same species at random. As before, at the beginning of the simulation the agents start with no prior knowledge and do not better than random. As time passes, the average recognition level over the past 50 time units is monitored, and increases up to a point where the agents in each species can recognize each other with a good accuracy.

Figure 11 shows the influence of space on the convergence to recognition patterns, as well as the influence of the negative learning and the transmission error. The negative learning does not have a significant effect on the agent performances. However space is found to be a major factor: When the search radius is too small the individuals from the same species do not learn to recognize each other as efficiently,



**Fig. 10.** Evolution of the recognition level vs. simulation time, when 2 symbols are in use (left), 3 symbols (middle), and 3 symbols with no transmission error (right).

	Small radius	Medium radius	Normal radius (Fig.10 middle)	No negative learning	No error (Fig. 10 right)
Mean Rec. Level	69.9	85.0	93.3	92.8	97.1
Dev. Rec. Level	10.2	6.96	4.76	4.70	2.75
Mean Num. Songs	13.1	11.6	11.4	10.5	4.31
Dev. Num. Songs	8.92	8.74	7.96	7.58	4.20

**Fig. 11.** Recognition levels and number of songs in each species at the end of the run, for various asynchronous scenario configurations, with 3 symbols and 6 species.

possibly due to the agents using different recognition patterns at different places, as is reflected by an increased number of songs. The effect of removing the 1% transmission error is however clearly visible: A better recognition rate, and much less diversity in the patterns used within a species. This contrasts from Fig. 5 in the synchronous case, where the number of songs was not noticeably affected by the removal of the transmission error.

## 7 Conclusion

A framework was presented where the genetic component and the knowledge acquired during an agent lifetime interact with each other: The genetic material defines the innate processing power of an individual, its capabilities for learning. In turn, the knowledge an agent acquires directly influences its success at reproduction. Both parts may be transmitted to the next generations: the genetic component using a crossover/mutation algorithm, and the knowledge using machine learning techniques built according to these genetic instructions.

The main findings of this work may be summarized by:

1. The learning mechanism needs to be simple and robust (failure of the linear model, Fig. 4 and 5, and of the ML classifier, Fig. 4).
2. Too powerful models are sufficient, but not necessary: the KNN model is simpler than the MLP and converges to the same performances (Fig. 4).
3. Complex recognition patterns are produced for free (Fig. 7, cycle lengths).
4. Asynchronous reproduction events in continuous time do not seem to alter the performances of the KNN model (Fig. 10).
5. However when the agents are too spatially isolated the recognition performance drops (Fig. 11).

The original problem of determining the minimal and sufficient conditions for the emergence of mutual recognition patterns can now be answered. According to the present study results, it seems that good candidate conditions are: 1. A turnover of agents in the genetic algorithm so as to produce new patterns, and 2. A limited form of knowledge transmission by imitating previous instances. In particular imperfect transmission is not a necessary condition (though no error improves performances), but a sufficient spatial distribution may be necessary. Additional experiments using more species (up to 12) and symbols (2,3,4) produce similar results that are not included here due to space restrictions.

An extension to this work could be the introduction of an external environment, allowing more advanced forms of communication, like stigmergy. The current setup has been restricted on purpose to a bare-bones model where the agents interactions

are strictly controlled. Yet, this prevents group effects and other collective behaviors that would be a natural extension to this framework. Another direction of research would be to investigate the influence of the learned part on the genetic component, the Baldwin effect [13]. Visual inspection suggests that in the current setup the genetic starter strings resemble the species specific song at the end of the training, but the more general question is why this is so and whether this is always necessarily the case. For example, for some AI architecture a genetic starter similar to the dominant species song introduces more training substrings, hence provide a selective advantage over individuals without the correct starter sequence. For more elaborated AI algorithms, and also for more complex environments with “social” interactions not restricted to choosing a mate based on its song, it is possible that the Baldwin effect operates in more complicated ways.

In any case, the current experiments have shown that very few preconditions are needed for the emergence of species-specific recognition patterns. What this shows is that the transition from isolated symbols to precursor sequences for more elaborated forms of communication, like language, is not exceptional. The more interesting question of how the precursor patterns may then turn or not into these advanced form of communication is, however, an open question.

Note: The code for the experiments described in this paper is available as free-libre software, GNU GPL license, on the author web page at <http://nicolas.brodu.free.fr>

## References

1. S.-Y. Wang, W., Ke, J., Minett W. J.: Computational Studies of Language Evolution. In Computational linguistics and Beyond. Huang, C.R. & Lenders, W. (2004) 65-106.
2. Wagner, K., Reggia, J. A., Uriagereka, J., Wilkinson, G. S.: Progress in the simulation of emergent communication and language, *Adaptive Behavior*, 11, 1, (2003) 37-69.
3. Harris, R. A.: Review of “The language instinct” by Stephen Pinker. *The Globe & Mail* (1994)
4. Smith, J. M., Szathmáry, E.: *The major transitions in evolution*. Oxford Univ. Press (1995).
5. Calvin W. H., Bickerton D.: *Lingua ex Machina: Reconciling Darwin and Chomsky with the human brain*. MIT press (2000)
6. Acerbi, A., Parisi, D.: Cultural Transmission Between and Within Generations. *J. of Artificial Societies and Social Simulation* 9(1)9, (2006).
7. Gong, T., Wang, S-Y. W.: Computational Modeling on Language Emergence: A Coevolution Model of Lexicon, Syntax, and Social Structure. *Language and Linguistics* 6, 1, (2005) 1-41.
8. Beecher D.M., Burt M.J.: The Role of Social Interaction in Bird Song Learning. In *Cur. Dir. in Psychological Science* 13, 6 (2004) 224-228.
9. Dessalles, J-L.: From protolanguage to language: model of a transition. *Marges linguistiques* 11, (2006) 142-152.
10. Slabbekoorn H., Smith T.B.: Bird song, ecology and speciation. In *Philosophical trans. of the Royal Society of London. Series B, Bio. Sciences.* 357,1420 (Apr. 2002):493-503.
11. Jackendoff R.: Possible stages in the evolution of the language capacity. *Trends in Cognitive Sciences* 3, 7, (July 1999).
12. Elliott, D.L.: A Better Activation Function for Artificial Neural Networks. Technical Report 93-8, Institute for Systems Research, University of Maryland (1993).
13. Baldwin, J. M.: A New Factor in Evolution. *American Naturalist* 30, (1896) 441-451, 536-553. Available online at [http://www.brocku.ca/MeadProject/Baldwin/Baldwin\\_1896\\_h.html](http://www.brocku.ca/MeadProject/Baldwin/Baldwin_1896_h.html)
14. Oudeyer P. Y., Kaplan F.: Language evolution as a Darwinian process: computational studies. *Cogn Process* (2007) 8:21–35.