Interacting with an artificial partner: modeling the role of emotional aspects

I. Cattinelli, M. Goldwurm, and N. A. Borghese
cattinelli@dsi.unimi.it

Dept. Computer Science, University of Milan, Italy

What is this about?

1. Introduction
   - History
   - Background
2. The basic model
   - Formulation
   - Implementation
3. The extended model
   - Formulation
   - Applying Reinforcement Learning
4. Results
5. Quantitative behavior analysis
   - Motivation
   - Markov chains theory
   - Markov chains for behavior analysis
   - Some examples
6. Conclusion
7. References

Probabilistic finite state automata

Reinforcement learning

Markov chains

It's been a long road...

- It all started with my Master Thesis (April 2006)
- The basic model proposed there has been successively extended and refined
- Finally, a quantitative analysis approach was developed based on Markov chains theory
- Published on *Biological Cybernetics* in 2008

Affective Computing

- What?
  - A fairly new interdisciplinary field, defined as *computing that relates to, arises from, or deliberately influences emotions* [1]
  - Contributions from Computer Science, Psychology, Neuroscience, ...
- Who?
  - Research in this field "officially" started in the 1990s with Rosalind Picard and her Affective Computing Group at MIT
  - In the last years the interest toward this research area has greatly grown, as proved by a number of dedicated conferences and workshops, papers and books
- How?
  - Implementation of modules for **human emotion recognition**, based on physiological parameters or on non-verbal communication
  - Design of systems for **simulating emotional states**, which can communicate emotions readable by the human user
  - **Models of emotional dynamics**, to explain how human emotional intelligence works and to reproduce this faculty in machines

Affective Computing

- ... and above all: Why???
  - To get truly intelligent machines: emotions are an important part of our intellective faculties!
  - To improve human-machine interaction, making it a bit closer to human-human interaction
  - Application domains: entertainment (video games, home robots), health care, social robots

The basic model

Let us consider a basic scenario where an artificial agent and a human partner interact.
The model for the agent's emotional dynamics is given by a four-tuple:

$$\langle S, U, P, s(0) \rangle$$

where:

- $S = \{s_1, s_2, \dots, s_N\}$ is the set of emotional states for the agent
- $U = \{u_1, u_2, \dots, u_M\}$ is the set of input (that is, the user's emotions)
- $P = \{P_0, P_1, \dots\}$ is the sequence of probabilistic transition functions:

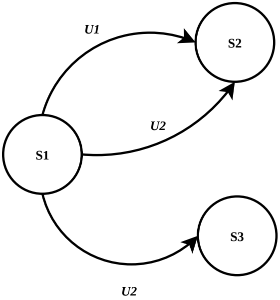
  $$P_t : S \times U \times S \to [0, 1] \text{ for } t = 0, 1, \dots$$

- $s(0)$ is the initial state.

The basic model

Therefore, our model is a **Probabilistic Finite State Automaton**...

Toy example:



P(S1, U1, S2) $= 1$
P(S1, U2, S2) $= 0.7$
P(S1, U2, S3) $= 0.3$

N.B.: $\sum_{s' \in S} P(s, u, s') = 1$ for each $(s, u) \in S \times U$

The basic model

As we said, our model is a **Probabilistic Finite State Automaton**... whose transition probabilities may change at each step.

So, how does it work?

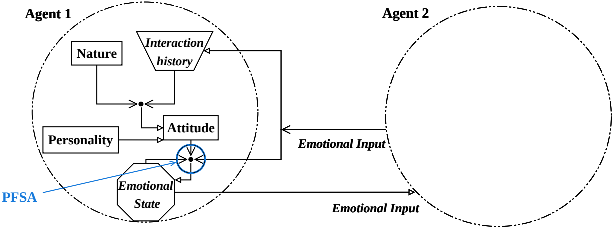
For each step t:

1. The agent receives the user's emotional state (e.g. by analyzing her facial expression);
2. Based on the agent's current state and input, P_t gives the probability of entering each possible next state;
3. A new emotional state is chosen by the agent based on these probabilities;
4. P_t is (possibly) modified to get P_{t+1};
5. Go to 1.

The basic model

We now introduce a specific terminology:

- The initial transition probability function, P_0, is called **personality** of the agent;
- The current transition probability function, P_t, is called **attitude** of the agent;
- The criterion that drives the update of the transition probabilities is called **nature** of the agent



The basic model

We have not mentioned, yet, how transition probabilities are being changed...

- Emotional inputs are grouped into K categories c_k (e.g. "nice" inputs)
- Each category has an eligibility trace $e_t(c_k)$ associated
- Each category has a set of *target states* $TS(c_k)$ associated
- When $e_t(c_k)$ exceeds a given threshold, the probability of entering the corresponding target states is incremented:

$$P_{t+1}(s, u, ts) = P_t(s, u, ts) + \Delta \quad \forall s \in S, u \in U, ts \in TS(c_k)$$

Target states for each category are defined by the agent's **nature**.
Example: for an imitative nature, $c_k =$ joyful inputs, $TS(c_k) = \{\text{JOYFUL}\}$

Reminder: Eligibility trace

The eligibility trace in TD(λ) algorithms keeps a history of visited states.

Here, the eligibility trace for each input category c_k keeps a history of received inputs:

$$e_t(c_k) = \begin{cases} \alpha e_{t-1}(c_k) + h(c_k, u_j) & \text{if the current input is clustered in category } c_k \\ \alpha e_{t-1}(c_k) & \text{otherwise} \end{cases}$$

- α is the decay parameter;
- $h(c_k, u_j)$ represents the affinity between the input and the category

Human-robot interaction

The basic model was at first implemented in a real human-robot interaction setting.

- Robot has 4 emotional states
  - NEUTRAL, JOYFUL, SAD, ANGRY
- User gives one of 7 emotional states as an input:
  - the six basic emotions according to Ekman [2] (JOYFUL, SAD, SURPRISED, ANGRY, FEARFUL, DISGUSTED), plus the NEUTRAL state
- Input is given via facial expressions, which are captured by the robot's camera and analyzed by basic image processing techniques
  - color segmentation, border extraction, block matching... → to get real-time processing
  - the facial expression is coded into a set of *Action Units* [3]
  - detected AUs are then mapped into emotions through a fuzzy-like scoring system

Video!

Agent-agent emotional interaction

Now, let us consider two synthetic agents interacting... How do we get there?
Simple! We use two PFSA:

$$A^1 = \left\langle S, U, P^1, s(0)^1 \right\rangle \text{ and } A^2 = \left\langle S, U, P^2, s(0)^2 \right\rangle \text{, where:}$$

- the set of emotional states S is the same for both A^1 and A^2;
- the set of possible inputs, U, is coincident with the possible states, S;
- the probabilistic transition functions, P_0^1 and P_0^2, are different at start, that is the two agents have different personalities;
- the initial states $s(0)^1$ and $s(0)^2$ are different.

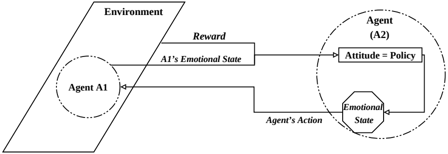
In brief: the state of A^1 is the input for A^2, and vice versa.

Learning Attitudes

Adaptation to the partner may be attained through the probabilities update mechanism described above...

... or, we can assign interaction goals to one agent and apply **reinforcement learning** [4]

- Agent A^1 acts as the environment, whose states
  - are observable by the learning agent
  - can be changed by the learning agent through its own "actions"
  - can be either goal or non-goal states
- Agent A^2 is the learning agent, and
  - receives positive reward when the environment gets to a goal state
  - has to learn a *policy* to maximize the long-term reward
- **Q-learning** [5] is used for optimal policy discovery



Reminder: Q-learning

Q-learning is a TD algorithm for learning the optimal action-value function $Q^*(s,a)$, which gives the expected return starting from s, executing the action a, and, from that on, following the optimal policy.

For every step of each learning episode, the function being learned, $Q(s,a)$, is updated according to

$$Q(s,a) = Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)] \tag{1}$$

Learning Attitudes

In this framework, $Q(s, a)$ is initialized to P_0^2.

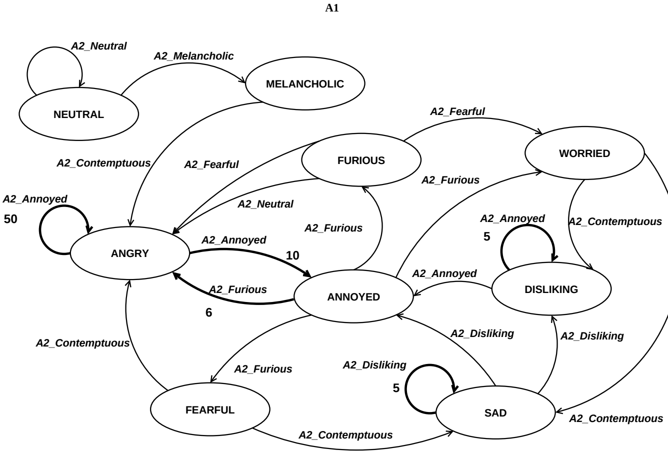
At each step t:

1. the learning agent observes state s and takes action a according to $Q(s, a)$: i.e., it takes action a, when seeing s, with a probability given by P_t^2;
2. the agent observes the new state s' and the associated reward ($= 1$ only if s' is a goal state);
3. Q ($= P_t^2$) is updated according to Eq. 1;
4. go to (1).

The policy being learned is therefore the agent's attitude.

Applying Reinforcement Learning: some results

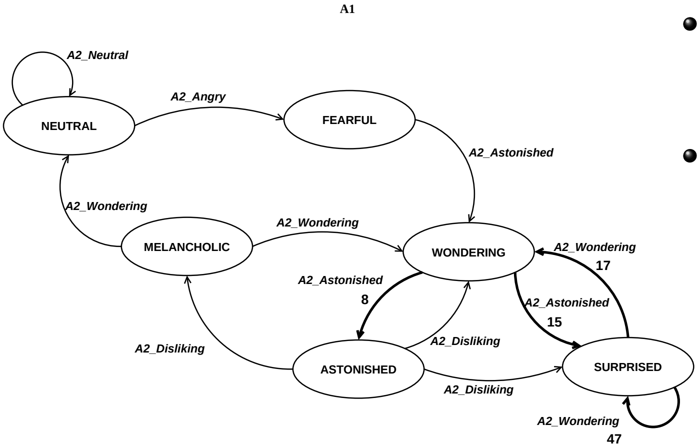
A^1 and A^2 start as "friendly" agents. Goal for A^2: making A^1 frequently **angry**



- Goal states = {ANNOYED, ANGRY, FURIOUS}
- Success rate on this instance of interaction: 78%

Applying Reinforcement Learning: some results

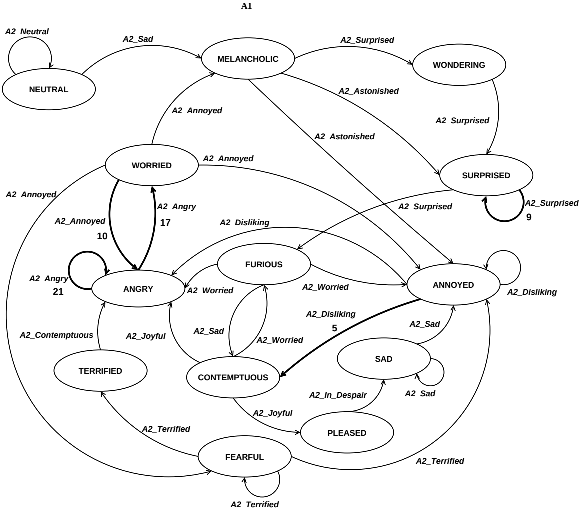
A^1 and A^2 start as "friendly" agents. Goal for A^2: making A^1 frequently **surprised**



- Goal states = {WONDERING, SURPRISED, ASTONISHED}
- Success rate on this instance of interaction: 95%

Applying Reinforcement Learning: some results

Let us go back to the 1st example: A^1 is "friendly", A^2's goal is to make it angry. A^2 has learnt the appropriate attitude... but now A^1's personality changes!



- Goal states = {ANNOYED, ANGRY, FURIOUS}
- Success rate on this instance of interaction: 51%

Quantitative behavior analysis

Problem: how can we evaluate such a model? Which quantitative measures can we derive?

Solution: Let us resort to Markov chains theory for a description of the asymptotic behavior of the system!

- Which states will be the most frequent ones?
- How long will it take to go from state i to state j?
- ...

Markov chains [6]

Given:

- a finite set of states, S;
- a probability distribution $\mu^{(0)}$ over S, termed the *initial distribution*
- a stochastic matrix P with indexes in S, called the *transition matrix*

Definition

a **finite homogeneous Markov chain** is a sequence of random variables $\{X_n\}_{n \in \mathbb{N}}$ such that

- for every $i \in S$, $\Pr(X_0 = i) = \mu^{(0)}(i)$
- for every integer $n > 0$, $i, j \in S$, and for every n-tuple $i_0, i_1, \ldots, i_{n-1}$, $\Pr(X_{n+1} = j | X_0 = i_0, X_1 = i_1, \ldots, X_{n-1} = i_{n-1}, X_n = i) = \Pr(X_{n+1} = j | X_n = i)$
- for every $n \in \mathbb{N}$ and $i, j \in S$, $\Pr(X_{n+1} = j | X_n = i) = p(i, j)$

Markov chains

Moreover, let us call $\mu^{(n)}$, for every integer n, the probability distribution of X_n. Then:

- $\Pr(X_n = j | X_0 = i) = (P^n)_{ij}$ → prob. of going from i to j in n steps
- $\mu_j^{(n)} = \Pr(X_n = j) = (\mu^{(0)\prime} P^n)_j$ → prob. of being in j at the n-th step

We are particularly interested in **primitive** Markov chains, that is chains having transition matrix P such that

$$P^k > 0 \text{ for some } k \in \mathbb{N}$$

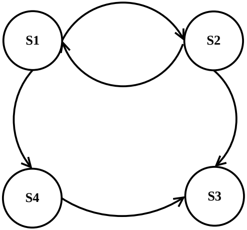
Markov chains

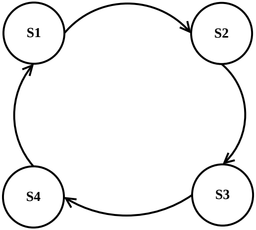
A primitive Markov chain is:

- irreducible → strongly connected transition graph
- aperiodic → the greatest common divisor of the lengths of cycles is 1

Question time!

1. Which graph is a strongly connected one? The one on the right!
2. Which one is aperiodic? None of these!

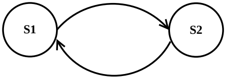




Markov chains

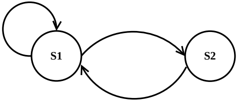
Let us recall that a primitive chain has a transition matrix P such that there exists a k for which $P^k > 0$.

Is this the same as requiring P to be irreducible?
No, aperiodicity is required too!



$$P = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$P^k = P$ for every odd k,
$P^k = I$ for every even k



$$P = \begin{bmatrix} 0.5 & 0.5 \\ 1 & 0 \end{bmatrix}$$

$$P^2 = \begin{bmatrix} 0.75 & 0.25 \\ 0.5 & 0.5 \end{bmatrix}$$

Properties of primitive Markov chains

1. There exists a unique **stationary distribution** π over S:

$$\pi' \; P \; = \pi'$$

   where π' is a left eigenvector of P corresponding to the eigenvalue 1

2. For every $i, j \in S$

$$\lim_{n \to +\infty} (P^n)_{ij} \; = \; \lim_{n \to +\infty} \mathsf{Pr}(X_n = j) \; = \; \pi_j$$

   that is, the limit distribution of X_n is independent from the initial state of the chain, and is coincident with the unique stationary distribution

Properties of primitive Markov chains

The error in the approximation of $\mu^{(n)}$ towards π can be kept arbitrarily small by controlling n.

3. For every $\varepsilon > 0$

$$d_{TV}(\mu^{(n)}, \pi) \ \leq \ \varepsilon$$

   for all $n \in \mathbb{N}$ such that

$$n \ \geq \ t\left(1 + \frac{\log_2 k - \log_2 \varepsilon - 1}{-\log_2 m(P^t)}\right)$$

   where

   - d_{TV} is the *total variation distance* between two probability distributions: $d_{TV}(\mu, \nu) \ = \ \frac{1}{2}\sum_{i \in S} |\mu_i - \nu_i|$
   - t is the smallest integer such that $P^t > 0$
   - k is the cardinality of S
   - $m(T)$ is a coefficient defined over a stochastic matrix T, such that $m(T) \ = \ \frac{1}{2}\max_{i,j \in S}\{\sum_{l \in S} |T_{il} - T_{jl}|\}$

Properties of primitive Markov chains - Average waiting time for first entrance

For every $j \in S$, let τ_j be the random variable defined by

$$\tau_j = \min\{n > 0 \mid X_n = j\}$$

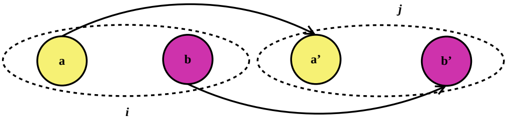
Then, $E_i(\tau_j) = E(\tau_j \mid X_0 = i)$ is the mean waiting time for the first entrance in j starting from state i.

4. $E_j(\tau_j) = 1/\pi_j$ for each $j \in S$
5. For $i \neq j$, the values $E_i(\tau_j)$ can be computed as well...
   - Let $G(z)$ be the matrix of polynomials in the variable z given by $G(z) = I - Pz$
   - Let $r_{ij}(z)$ be the entry of indexes i, j of the adjunct of $G(z)$: $r_{ij}(z) = (-1)^{i+j}\text{det}(G_{ji}(z))$ where $G_{ji}(z)$ is the matrix obtained from $G(z)$ by deleting the j-th row and the i-th column
   - $E_i(\tau_j) = \frac{r'_{ij}r_{jj} - r_{ij}r'_{jj}}{r^2_{jj}}$, where $r_{ij} = r_{ij}(1)$, $r_{jj} = r_{jj}(1)$, $r'_{ij} = r'_{ij}(1)$ and $r'_{jj} = r'_{jj}(1)$

Markov chains and the interaction model

How can all this be related to our model?
Markov chains **have no inputs**!
Yes, but...

- We can build one transition matrix, M, for the whole interaction system
- $M(i,j)$ gives the probability to go from state $i = (a,b)$ to state $j = (a',b')$, with a, a' emotional states for agent A^1, and b, b' states for A^2



- $M(i,j) = P^1(a,b,a') \times P^2(b,a',b')$

... and so now we have all the ingredients for a Markov chain!

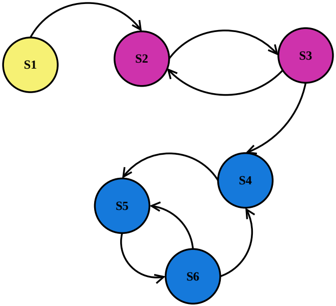
Markov chains and the interaction model

We have seen that primitive Markov chains have interesting properties, so: is our M primitive?
No! Because it is generally not irreducible...
Solution: let us reduce it!

- M not irreducible $\rightarrow$ the transition graph has more than one strongly connected component
- Some of them will be *essential components*: once entered, they will never be left

Markov chains and the interaction model

Example of a reducible chain



We have three strongly connected component, with just the blue one being essential (and aperiodic, too).

We can reduce our system just to the blue states, and what we get is a primitive chain.

In our examples, M turns out to have only one essential (and aperiodic) component $\rightarrow$ this M_{red} is primitive, and we can apply the above properties!

Quantitative behavior analysis – Limit probability of states

Let us consider again the previously shown interaction systems:

1. A^1 friendly, A^2 acquired a policy for making the partner angry most of the time (fig.)
   - M_{red} is composed of 15 states
   - the most probable states according to π are
     - (ANGRY, ANNOYED), with $p = 0.5148$
     - (ANNOYED, FURIOUS), with $p = 0.1548$
     - (SAD, DISLIKING), with $p = 0.0973$
2. A^1 friendly, A^2 acquired a policy for making the partner surprised most of the time (fig.)
   - M_{red} is composed of 10 states
   - the most probable states according to π are
     - (SURPRISED, WONDERING), with $p = 0.6286$
     - (WONDERING, ASTONISHED), with $p = 0.2292$
     - (ASTONISHED, DISLIKING), with $p = 0.0917$

Quantitative behavior analysis – Limit probability of states

What does this analysis tell us?

- Probability values provided by the stationary distribution are rather close to the frequencies observed in the experiments
  - the stationary distribution is a suitable descriptor of the actual behavior of the systems even after a limited amount of steps
  - the error in approximation is less than 0.001 just after 38 and 27 steps, respectively (see Prop. 3)
- The reinforcement learning process was effective
  - the goal states defined for A^1 are among the most probable states of the system in each of the considered examples

Quantitative behavior analysis – Mean entrance times

We can define a set of *starting states*, SS, and a set of *ending states*, ES, and use Prop. 4–5 to compute the mean entrance times for going from states in SS to states in ES.

Natural choice in a learning scenario: ES coincident with goal states...

1. A^1 friendly, A^2 acquired a policy for making the partner angry most of the time
   - $ES = \{(a, b) \mid a = \{\text{ANNOYED}, \text{ANGRY}, \text{FURIOUS}\},\ b \in S\}$
   - $SS = \{(\text{MELANCHOLIC}, \text{CONTEMPTUOUS})\}$
   - a minimum of 5.91 and a maximum of 213.10 steps, on average, for going from states in SS to states in ES (mean 77.98)
2. A^1 friendly, A^2 acquired a policy for making the partner surprised most of the time
   - $ES = \{(a, b) \mid a = \{\text{WONDERING, SURPRISED, ASTONISHED}\},\ b \in S\}$
   - $SS = \{(\text{NEUTRAL}, \text{ANGRY})\}$
   - a minimum of 3.86 and a maximum of 12.43 steps, on average, for going from states in SS to states in ES (mean 7.07)

Quantitative behavior analysis – Mean entrance times

What does this analysis tell us?

- In the second example, the learned policy is particularly effective in driving A^1's behavior to the given goals
  - just 7 steps are required, on average, to reach a goal state!
- In the first example, the policy is less effective, meaning that about 78 steps are required, on average, to reach a goal state...
  - ... however this is mainly due to two particular end states that have very low entrance probabilities
  - the other three goal states can be reached within 30 steps

Summing up

We proposed an emotional interaction model:

- for a human-robot, or for an agent-agent interactions scenario
- having a probabilistic and time-varying nature, leading to more life-like interactions
- capable of adaptation to the interlocutor, either by the probabilities update mechanism or by autonomous learning
- with a basic structure that can easily be extended (adding/modifying states, inputs, personalities, ...)
- which can be employed, for instance, as a basis for emotional agents in video games, or in social robotics

Summing up

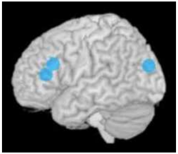
We showed how a quantitative analysis of interaction scenarios can be attained by resorting to Markov chains theory:

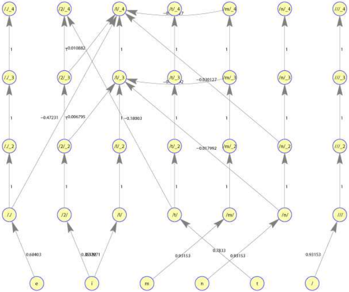
- the stationary distribution of the system highlights its most probable states, largely corresponding to goal states as defined in the reinforcement learning framework
- mean waiting times are used to establish the number of steps required, on average, for the system to reach a set of states of interest (e.g. goal states)

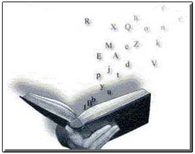
Therefore, this analysis can provide a measure of the effectiveness of learned policies

What's next?

Currently, I am moving to different topics...







... but there's still work to do on this topic (*read*: Theses available)

Highlight: Markov chain-based analysis in the non-stationary case

References

R. Picard.
Affective Computing.
MIT Press, 1997.

P. Ekman.
An argument for basic emotions.
Cognition and Emotion, 6(3–4): 169–200, 1992.

P. Ekman and W. V. Friesen.
Manual for the Facial Action Coding System.
Consulting Psychologists Press, Inc., 1978.

R. Sutton and A. Barto.
Reinforcement Learning: An Introduction.
The MIT Press, 1998.

C. Watkins.
Learning from Delayed Rewards.
PhD Thesis, University of Cambridge, England, 1989.

M. Iosifescu.
Finite Markov Processes and Their Applications.
Wiley, 1980.