Learning and Evolving Agents in User Monitoring and Training

Stefania Costantini *Pierangelo Dell'Acqua*
Luís Moniz Pereira *Francesca Toni*

Accompanying paper:
http://centria.di.fct.unl.pt/~lmp/publications/online-papers/AICA-2010.pdf

Abstract

- We propose a general vision for agents in Ambient Intelligent applications, where they monitor and unintrusively train human users.
- And learn their patterns of behavior, not just by observing and generalizing their observations, but also by “imitating” them.
- Agents can learn by “imitating” other agents too, by “being told” what to do.
- In this vision, agents collectively need to evolve, and together take into account what they learn from, or about users, as a result of monitoring them.

Intro and Motivation - 1

We supply a framework for agents to improve the "quality of life" of users, by efficiently supporting their activities.

- Aiming to monitor them to ensure a degree of coherence in behavior.
- Training them at some task.

And bring advantages to users, in they being:

- Relieved of some behavioral responsibilities, e.g. directions on the "right thing" to do.
- Assisted when they perceive themselves partly inadequate for a task.
- Told how to cope with unknown, unwanted, or challenging circumstances.
- Helped by a "Personal Assistant" improving in time its understanding of user needs, cultural level, preferred explanations, its coping with the environment, etc.

Intro and Motivation - 2

Agents are able to:

- Elicit, by learning, behavioral patterns the user is adopting.
- Learn rules and plans from other agents by imitation (by "being told").

We are inspired by evolutionary cultural studies of human societal organization to collectively cope with their environment. Principles emerging from these studies equally apply to societies of agents.

Especially if agents cooperate helping humans adapt to new environments and/or when the ability to cope is too costly, non-existent or impaired.

Agents modify or reinforce rules/plans/patterns they hold, based on an evaluation performed by an internal meta-control component. Evaluation leads agents to modify behavior via their evolving abilities.

The model accords with Ambient Intelligence as a digitally augmented human centered environment, where appliances and services proactively and unintrusively provide assistance.

Innovation and Imitation - 1

We consider it necessary for an agent to acquire knowledge from other agents, i.e. learn "by being told" instead of learning only by experience.

Indeed, this is a fairly practical and economical way of increasing abilities, widely used by human beings, and widely studied in evolutionary biology.

Avoiding the costs of learning is an important benefit of imitation. An agent that learns and re-elaborates the learnt knowledge becomes in turn an information producer, from which others learn in turn.

On the other hand, an agent that just imitates blindly can be a burden for the society to which it belongs.

Innovation and Imitation - 2

Evolutionary biology shows the long-run of evolution of human societies is a mixture of learners and copiers, where both types have the same fitness as would purely individual learners in a population without copiers.

To understand this, think of imitators as information scroungers and of learners as information producers.

Information producers bear a cost to learn. When scroungers are rare and producers common, almost all scroungers will imitate a producer. If the environment changes, any scroungers imitating scroungers will get caught out with bad information, whereas producers will adapt.

Thus, an agent is able to increase its fitness in such a society in 2 ways:

- If it is capable of usefully exploiting learnable knowledge, hence deriving new knowledge and becoming an information producer.
- If it is capable to learn selectively, learning when learning is cheap and accurate, and imitating otherwise.

Innovation and Imitation - 3

We outline a model so inspired, for the construction of logical agents able to learn and adapt their behavior in interaction with humans.

We emphasize that, to engage with humans, agents should have a description of how humans normally function.

The starting description limited to "normal" user behavior in some ambient setting. Agents are deliberately designed and originally primed with the ambient setting in mind, and humans are new to the setting and/or experience difficulties or impairments in coping with it.

As deep learning (i.e. learning from scratch) is time consuming and costly, it needs not be repeated by one and all, so an agent may apply a hybrid combination of deep learning and imitation.

The view is that all agents and the society as a whole benefit from the learning/imitation process, envisaged as a form of cooperation.

Innovation and Imitation - 4

Each agent is initially equipped either with sibling agents or with a structured agent society having abilities related to its "role", i.e., with the supervision task it will perform.

Initial capabilities may be enhanced by internal learning, consequence of interaction with user, environment, and similar agents.

When some piece of knowledge is missing, and a task cannot be properly carried out by an agent, that piece may be acquired from the society, if extant there, and if the agent is unable or unwilling to deep learn it.

Next, it will exercise it in the context at hand, subsequently evaluate it on the basis of experience, and report back to the society.

The evaluation of imparted knowledge builds up a network of agents' credibility and trustworthiness, where the learning producers benefit from the more extensive testing performed by scroungers.

Multi-layer Monitoring - 1

A flexible interaction with the user is made easier by adopting a multi-layered agent model, where there is a base level, called PA for "Personal Assistant", and one (or more) meta-layers, called MPA.

While the PA is responsible for the direct interaction with the user, the MPA is responsible for correct and timely PA behavior.

Thus, while the PA monitors the user, the MPA monitors the PA. The actions the PA undertakes include, for instance, behavioral suggestions, appliance manipulation, enabling or disabling user manipulation of an appliance.

The actions the MPA undertakes include modification of the PA in terms of adding/removing knowledge (modules) in the attempt at correcting inadequacies and generating more appropriate behavior.

Multi-layer Monitoring - 2

In our framework, both the PA and the MPA will largely base their behavior upon verification of temporal-logic rules that describe expected and un-expected/unwanted situations.

Whenever all rules are complied with, the overall agent is supposed to work well.

Whenever some rule is violated, suitable actions are to be undertaken, to restore correct functioning.

Temporal rules are checked at run-time –at a certain frequency and with certain priorities– and necessary actions are then executed.

Learning, evolution, self-checking - 1

Agents act not in isolation, being part of a society: in its simplest form, one of sibling agents. Generally, it may be a structured society of agents sharing common knowledge and goals.

Assume agents in this society are benevolent and willing to cooperate, or have evolved to become so.

Agents monitoring/training a user must treat at least 3 kinds of learning activities:

Initialization: to start its monitoring/training activities, an agent receives from a sibling or society basic facts and rules defining:

- the role it will impersonate with the user
- the basic behavior of the agent

This is clearly a form of *learning by being told*.

Learning, evolution, self-checking - 2

Observation: an agent observes the user's behavior along time in different situations.

It collects observations and classifies them to elicit general rules, or at least become able to expect with reasonable confidence what the user will do in future.

Interaction: whenever the monitoring/training agent has to cope with a situation for which it has no sufficient knowledge/expertise, it tries to obtain, from other agents or from the society, the necessary knowledge and rules.

The agent will in general evaluate the actual usefulness of the so acquired knowledge.

Temporal-logic meta-rules - 1

Included in the initialization stage are general temporal-logic meta-rules, included in the MPA.

The two interval temporal logic rules below state the user should eventually perform necessary actions within the associated time-threshold. And should never perform forbidden actions:

FINALLY *(T) A :: action(A), mandatory(user, A), timeout(A, T)*

NEVER *A :: action(A), forbidden(user, A)*

These meta-rules are checked dynamically, i.e. at run-time, at a certain (customizable) frequency.

Meta-rules can themselves be customized by the agent, through learning, after a relevant number of interactions with a user.

Temporal-logic meta-rules - 2

Assume an agent is required to act as a baby-sitter. The knowledge it will be equipped with can include the following.

A mandatory rule states children should always go to bed within a certain time period:

ALWAYS *go_to_bed(P, T), early(T) :: child(P)*

The agent may later learn, through observations, what "early" means according to childrens' age and family habits, and elicit a rule such as:

USUALLY *go_to_bed(P, T), 9:00 ≤ T ≤ 10:30 :: child (P), age (P, E), 10≤ E ≤13*

Vice versa, each agent contributes to the society. This rule can be communicated to the society and –after suitable evaluation by the society itself– be integrated into its common knowledge and communicated to other agents.

Common Belief Set - 1

An agent may contribute to the society's "common belief set" under several respects:

- Provide others with its own knowledge when required.
- In a structured society, insert into a repository whatever it has learnt.
- Provide feedback on the usefulness/effectiveness, within its own context, of the knowledge it has been told by others.
- Participate in "collective evaluations" of learnt knowledge.

Facts and rules that a monitoring/training agent learns from the interaction with the user can be very important for the society, in that they can constitute knowledge agents may acquire "by being told".

An agent can later on verify the adequacy of learnt rules, and promptly revise/retract them in face of new evidence.

Common Belief Set - 2

Hopefully, after some iterations of this building/refinement cycle, the built knowledge is "good enough" in the sense the predictions it makes are accurate "enough" concerning the environment observations obtained with experience.

At this point, the theory can be used both to explain observations and produce new predictions. In computational logic, several approaches to learning rules and facts have been developed.

In real-world problems, complete information about the world is impossible to achieve, and it is necessary to reason and act on the basis of the available partial information and hypotheticals.

In situations of incomplete knowledge, it is important to distinguish between what is true, what is false, and what is unknown or undefined.

Common Belief Set - 3

After a theory has been built, it can be exploited, on the one hand to analyze observations and provide explanations for them; on the other, to foretell user behavior.

Note that in practical situations several possible alternative rules might be learnt. The MPA should include suitable Integrity Constraints (ICs) and preferences for choosing amongst alternatives.

Moreover, the learnt rules should be compared with subsequent observations, and thence might be refined, revised or dropped. In this matter, the role of the society can be crucial.

Evolutionary Inspiration - 1

Finding possible alternative explanations is one problem; finding the "best" another issue altogether.

One may assume "best" means minimal set of hypotheses, and we describe a method to find such a "best".

Another interpretation of "best" is "most probable", and in this case the theory inside the agents must contain adequate probabilistic information.

Ex contradictione quodlibet. This well-known Latin saying means "Anything follows from a contradiction". But contradictory, oppositional ideas and arguments can be combined together in different ways to produce new ideas.

Because "anything follows from contradiction", a thing that might follow is a solution to a problem to which several alternative positions contribute.

Evolutionary Inspiration - 2

A well-known method for solving complex problems widely used by creative teams is 'brainstorming'. In a nutshell, every agent participating in a 'brainstorm' contributes by adding ideas to an 'idea-pool' shared by the agents.

All the ideas, sometimes clashing and oppositional among each other, are then mixed, crossed and mutated. The solution to a problem can arise from the pool after so many iterations of a selective evolutionary process.

The evolution of alternative ideas and arguments, in order to find a collaborative solution to a group problem, is one underlying inspiration of our work.

Evolutionary Inspiration - 3

Darwin's theory is based on natural selection: only individuals better fit for their environment survive, and generate new offspring by reproduction. Individuals also suffer random mutations of genes that transmit to offspring.

Lamarck's theory in contrast states evolution is due to a process of environment adaptation individuals perform in lifetime. The result of this process being transmitted to offspring via the genes. This, however, is not physiologically true.

But Lamarckian evolution has received renewed attention, since it can model cultural evolution. Thence the concept of "meme" was developed, a cognitive equivalent of 'gene', storing lifetime abilities learnt by individuals or groups, and culturally transmitted to offspring.

In genetic programming Lamarckian evolution is a powerful concept.

User Monitoring and Training - 1

The next scenario illustrates dynamic aspects of the KB of a PA/MPA whose knowledge evolves to reflect changes in user behavior and environment.

Suppose a user must undergo treatment for some illness and therefore take medicine. She asks her personal assistant about what to do during treatment, e.g., "Can I drink a glass of wine if I have to take this medicine?"

More generally, the user may just ask "Can I drink a glass of wine now?" and the PA should give advice based on whether there is medicine to be taken (or other related matters).

As discussed before, the agent and its PA will have been equipped by the society with initial knowledge about its task. However, if the available knowledge turns out to be either missing or inadequate, then the PA is able to resort to the MPA.

User Monitoring and Training - 2

User asks: "Can I drink a glass of wine now?" and the agent finds no answer in its present beliefs state. The PA might be equipped with rule:

ALWAYS *asks(user, do(action, A)), known(A) ÷ lookup (A)*

If this rule is not enacted, and it can only be because action A is not known, then the agent attempts to discover what A is with *lookup(A)*.

The corresponding reactive rule in MPA and might be:

lookup (A) ← check(A)

check (A) ← found_module (A, M), assert (M)

check (A) ← not found_module(A, M), learn (A, M), assert (M)

The reactive rule performs *check(A)*: if it finds in MPA a module M coping with A, then M is added (asserted) in PA; else, MPA triggers a learning process –*learn(A, M)*– returning a module to be asserted.

Learning is "by being told": MPA will obtain M from the society.

User Monitoring and Training - 3

PA will not contain the plain constraint that one should not drink alcohol and take medicine:

$\perp \leftarrow$ drink, take medicine

as it provides no temporal information for returning a reliable answer.

Rather, it may contain the A-ILTL rule stating that one should never drink alcohol within sixty minutes before or after the consumption of medicine:

NEVER *(drink: T1), (take_medicine: T2), T1−T2 < 60*

The rule can be exploited both to block an action, if the other one has been performed already, or to provide explanations, should the user ask for advice.

User Monitoring and Training - 4

If the user is training taking medicine, we may define a rule stating which medicine to take before dinner. Towards getting trained, the user tells the system which actions she is about to do.

ALWAYS *(take_medicine(M) : T1), (have dinner: T2), T1−T2 < 30 ::*
dinnertime(T1), indication(M, beforedinner) ÷ train_user_md

train_user_md ← ...

The *ALWAYS* rule is false if one conjunct is: if *train_user_md* it must be checked whether dinner-time is near and appropriate to take medicine, or if user is going to have dinner but forgot the medicine required taking before dinner.

Modifying its behavior, the system checks context to tell user what to do when. It may control treatment is effective by checking if user has recovered after a certain time (say, 1 week). Else, treatment is revised.

FINALLY *(T) recovered (T) :: T = 1week ÷ revise_treatment*

Future Work

We aim at designing the meta-meta level for controlling knowledge exchange.

Particular attention should be dedicated to strategies involving reputation and trust for the evaluation of learnt knowledge.