Helpful Robots are Clever Robots

Mary Felkin
CADIA
Center for Analysis and Design
of Intelligent Agents
Reykjavik University
mary@ru.is

Yves Kodratoff
LRI
Laboratoire de Recherche
en Informatique
Université Paris Sud
yvko@free.fr

ABSTRACT

We developed a system to obtain a robot able to learn to emulate the strategies used by a human facing a problem-solving task. We have been able to solve this problem within a particular setting in which the human behaviour can be interpreted as time series of 'observables' of his/her problem-solving strategy. Our solution encompasses the solution of yet another problem, namely how to close a loop starting with the behaviour of several humans, then going through its analysis all the way up to the human strategies, automatically transferring these strategies to a humanoid robot and finally running this robot to generate observable data of the same type as the ones obtained from observing the humans.

1. INTRODUCTION

As early as 1984, Michael Brady placed robotics as an important test field for artificial intelligence algorithms [5]. But the work on intelligent robots suffered for a while from a lack of interest from industrialists. Special purpose industrial robots could perform their productive task without fuss, and the appearance of the slow-walking Honda's Asimo did nothing to convince industrialists of the usefulness of humanoids. But killer applications exist for humanoid robots. Their advantage is that they function in the real world and will be able to accomplish many of the tedious tasks of everyday life, such as cleaning the house, making the beds, driving the kids to school, making sure granny takes her medication, etc. All the tools, cooking appliances, etc. are designed to be used by humans, so if a robot is to make itself useful in a house it will have to be humanoid. As the general population grows older, the need for close medical attention and general care will increase.

In this context, the need for efficient human-robot interfaces is crucial. It seems very likely that the average robot user will no more be willing to program his/her robot then the average PC user is willing to program his/her PC. Learning by imitation provides an intuitive way for users to teach new tasks to their robots, because this is how a human would teach another human. As each household is different, the number of possible tasks in the real world is huge. Many people also hold strong views concerning the best way to perform household chores. Our system could enable robots to learn, not just to cook an egg, but to cook it like its user/teacher would cook it, and user satisfaction will play an important role. Robots are not yet sufficiently advanced to test whether people will expect a more intelligent behaviour from a humanoid robot than from a box-shaped computer, but even if the humanoid shape does not influence them into increasing their expectations in this regard, realtime obedience in the real world is safety-critical and a robot ignoring a "Stop!" command because it is busy installing an update could lead to disaster.

Intelligent behaviour in unpredictable real-life environments can only be achieved through endowing the robot with learning abilities. Even non humanoid robots would benefit from them. A robot vacuum cleaner could be a tripping hazard to a visually impaired person, and making them audibly signal their presence would be annoying. Better to have the robot quietly move away from the person and going to hide in an out-of-the-way spot. But a young child or a persistently playful pet could then prevent the robot from doing its job by chasing it. A user-interface enabling the owner to order the robot to "Get out of the way of my mother but ignore my son" is required to solve this problem and countless others, and this in turn requires powerful learning algorithms to be running in the background.

The type of learning know as plan and/or goal recognition [11], which our work takes one step further into strategy recognition, will also have an important role to play. It solves correspondence problems[1]. It is notoriously difficult for a humanoid robot to carry a full cup of coffee, because it implies very fine movement coordination, but a robot which recognises the purpose can bring the empty cup and the coffee pot, and only fill the cup once the destination has been reached. A robot which recognises the strategy can do this even in previously unknown surroundings.

2. RELATED WORKS

A large amount of work has been done in the field of robot learning by imitation, a relatively new (about twenty years

.

[1] In robotics, "correspondence problems" refers to the problems which arise from the fact that no (existing) robot is built exactly like a human. A humanoid robot may have a morphology and physical capacities which are quite different from these of its instructor [2].

old) field of research, see for example [4], [9] and [12]. This field takes inspiration from a wide range of disciplines, including psychology, biology, neurobiology, etc. [1], [3], [8] and [6]. An example among others of the necessary multidisciplinarity is [2] who propose a mathematical solution to the correspondence problem, which originally comes from animal psychology: they formalise the correspondences by giving mapping matrices to link agents with different morphologies. Other research papers present work which is less biomimetic, for example [7] who present an architecture for extracting the relevant features of a given task and then generalise the acquired knowledge to other contexts. They demonstrated the effectiveness of their architecture by implementing it on a humanoid robot learning to reproduce the gestures of a human teacher. A formal definition of plan recognition can be found in [11]. We take goal and intention recognition a step further and we close the data-robot-data learning loop so the robot which learns from a human instructor can later become the teacher to another robot (or even to a human).

3. EXPERIMENTAL SETTINGS

In a sequence of psychological experiments, blindfolded human volunteers explored a maze[2] in search of a treasure and, while doing so, expressed their search strategy[3] by sequences of perception-actions pairs, which were filmed. Perception here was limited to touch, and actions were limited to moving in the maze, touching objects and picking up the treasure, all of which could be observed on the videos. The psychologist [10] and the mixed team [14] showed that the volunteers in the mazes had several different goals which they combined through some thought process akin to multicriteria optimisation to mentally construct and evaluate their behaviours. On top of their given goal, finding the treasure, their most often used strategies included the goals of not getting lost, of not exploring the same place twice, of not bumping into obstacles, etc.

4. STEPS

We performed a detailed analysis of the videos, starting with a digitalisation, and including feature extraction, automated feature selection and semi-automated tracking. Our goal was to learn the underlying strategy of each volunteer and to abstract it sufficiently reproduce it in new contexts.

Automatically extracting from a database the strategies used by humans in a problem-solving situation takes more than a good preprocessing and then running the database through the appropriate data mining algorithm. To go from the database of observables to the strategies, we had to define a middle ground. Fig.1 (left) models the human's cognitive processes as a very simplified version of the HCog-Aff (Human Cognitive Affects, [13]) model, and superimposes our definitions. The raw data, called the observables, are indicated in fig.1. They are the basic facts such as the position of the body of the person in the maze at a given time step, etc. Primitives are combinations of observables, and sometimes of observables and static maze descriptors. Tactics are a combination of observables and primitives. We defined 4 categories of tactics and the combined effect of enacting one of each is a strategy. Fig.1 shows the path we followed in this work: First a bottom-up generalisation, in several steps, which started with the log file recording the movements of the human and was achieved with the help of machine learning algorithms. Then the top-down implementation of the induced strategies into control variables and (robot) body movements.

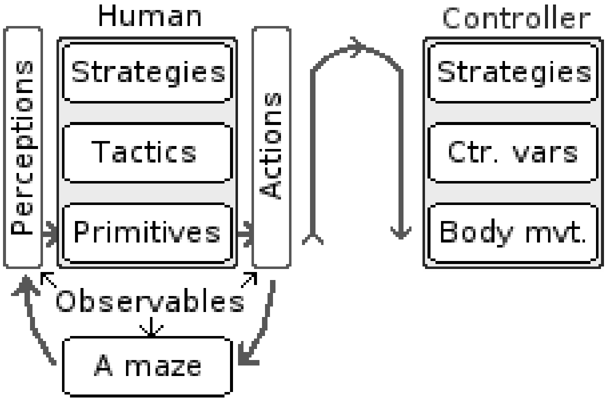

Figure 1: From human strategies to robot strategies

5. STRATEGIES

A situation variable is a descriptor of perceptions of the environment external to the controller[4]. Each of the M situation variables has a finite and known number of possible values. A control variable is a descriptor of robot action. Each of the N control variables has a finite and known number of possible values.

Formally, a robotic strategy is:

- A finite set of external situation states, E. Each situation state of E is expressed by a vector of M situation variables values: $(e_1, ..., e_M)$.
- A finite set of internal action states, I. Each action state of I is expressed by a vector of N control variables values: $(i_1, ..., i_N)$.
- An action transition matrix mapping all possible situation states to all possible action states. The values contained in this matrix are the probabilities of the robot enacting the behaviour described by an action state given a situation. We call it $\Lambda_A = a_{ij}$.
- An action duration mean transition matrix mapping all possible situation states to all possible action states. The values contained in this matrix are the means, should the robot enact the behaviour described by an action state, of the duration of all control variables of this behaviour. We call it $\Lambda_{AD} = a_{d-ij}$.
- An action duration standard deviation transition matrix mapping all possible situation states to all possible

[2] The mazes were not virtual, they were built with rows of tables and sometimes cupboards in a large room.

[3] Our use of the expression "search strategy" here does not imply the volunteers were searching according to an explicit plan. Randomly searching through the maze is also a "search strategy", and so is "not searching at all".

[4] Not necessarily "external to the robot", the input from a sensor describing the state of the internal battery would be a situation variable value.

action states. The values contained in this matrix are the standard deviations, should the robot enact the behaviour described by an action state, of the duration of all control variables of this behaviour. We call it $\Lambda_{ASD} = a_{sd-ij}$.

Whenever the situation state of the robot changes, the robot goes into a certain action state chosen randomly according to the probabilities of Λ_A. It draws durations, in independent draws, for all the control variables values according to the Gaussian probability distributions defined by Λ_{AD} and Λ_{ASD} and starts a count down to implement these durations. When the situation state of the robot does not change but one of the control variable values reaches the end of its randomly assigned number of time steps, the robot goes into another action state chosen randomly, according to the probabilities of Λ_A, among all action states which have the same values for all the other control variables and a different value for the control variable which is due for a change. It draws a duration for this new control variable value according to the probability distributions defined by Λ_{AD} and Λ_{ASD}. So switches from one action state to another can be triggered both externally, by a change of situation state, and internally, by the reaching the end of some control variable value life span. When the log file shows a such a switch happening independently of these two conditions, it corresponds to a change of strategy. Changes of strategy are defined by a subset of situation states, either of which triggers the change, by a consecutive sequence of situation states belonging to this subset, or by a time limit assigned to each consecutive strategy. When a strategy is reproduced from a given log file, the probabilities of transitions which never occurred can be set to zero and the corresponding values of Λ_{AD} and Λ_{ASD} left undefined.

This definition of a strategy resembles a Hidden Markov Model (HMM). Our Λ_A corresponds to the confusion matrix of a HMM. But we have no internal state transition matrix, it is replaced by two external matrices, grounding every internal transition probability into the external context and time. This also removes the need for the vector of initial internal state probabilities. Another difference is that the count down mechanism makes the internal state of the system dependant not only upon the previous internal state, nor upon any fixed number of previous states, but upon a variable number of previous states. Our two external matrices Λ_{AD} and Λ_{ASD} models the variability over time intrinsic to human behaviour. In every day speech, a ”robotic behaviour” has come to mean an inhumanly rigid and repeating behaviour. Our goal being to learn human problem-solving strategies our model needed this flexibility over time. Moreover, we believe that this flexibility is one of the advantages humans have over robots, an advantage which contributes to making humans more efficient in real-life situations. Λ_{AD} and Λ_{ASD} also have “the opposite effect” as they enable the robot to “remember what it is doing” and so preclude erratic behaviour.

Brute force mimicking of a human strategy according to this definition of a strategy would be trivial (the matrices can be filled by counting the relevant occurrences in the log file corresponding to the strategy of this human) but it would also be intractable in any but the most basic settings and it would require very large log files. Luckily strong simplifications are possible without deterring from human-like behaviour. In a complex situation these simplifications cannot be hand-crafted, they can only be achieved through the use of machine learning algorithms. Our definition has an inherent simplicity in that it only takes into account the influences of situation variables upon action variables. The influences of action variables upon situation variables are not part of our definition of a strategy. The influences of action variables upon situation variables describe the maze and not the strategy. One simplification is to limit the length of the vectors of E, another is to eliminate all task-achievement-irrelevant behaviours (even if the psychologists find them meaningful). Attribute construction and attribute selection algorithms were used to build and select useful descriptors.

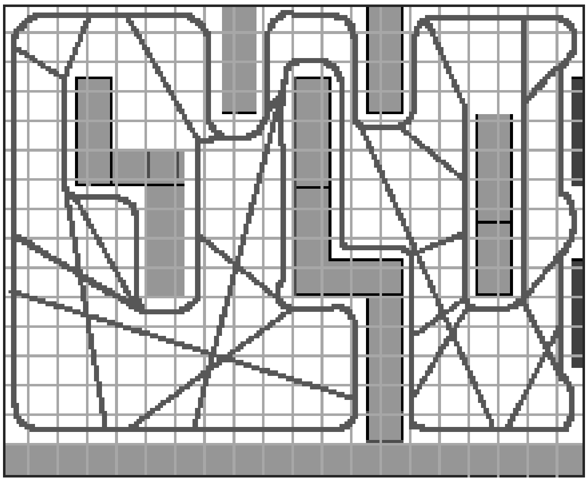

Figure 2: 10 minutes robot run in a maze

6. RESULTS

We consider that the fact that some human problem-solving strategies are learnable is more important than the actual strategies being learnt here.

The purpose of the volunteers was to find the treasure, so their performance in terms of time to achievement and/or ground coverage depended too much upon the original location of the treasure for performance comparisons to be made. We only had the logs of ten human runs. Each human only went through a maze once. They did not all go through the same maze. Three humans definitely did not spend all their time in the maze searching for the treasure. The logs of “performant humans” do not generate better controllers. In fact in one case it was quite the opposite. On average the humans took 11’32” to find the treasure. Our controller induced from the combined logs of all humans reached between 78% and 94% of all reachable squares (ground and tables) at least once after 10 minutes, depending on maze size and complexity. But concluding that, given uniformly random treasure locations, the robot did better than the humans would be, to say the least, premature. Given these facts and lack of human-related data we regretfully forwent comparison measurements.

Our program can read the log file of any individual human volunteer and automatically translate it into an implemented robotic strategy. But the results we give here are the results of the program generated from all logs joined together. Fig.2 records the track of such a run of our program in a maze. The unreached squares, about 14%, are the ones which are part of the (white) ground and not touched by the track. We could also have unreached squares on the tables, though this is not the case here: the robot went at least once along a side (or both) of all table-made "maze walls" and explored them all (the track shows the position of the body of the robot, not the position of the hands). The settings for the following were the four mazes from which one or several logs had been drawn, and six extra (invented) mazes used for testing purposes.

- The performance was not better in the "known" mazes than in the invented mazes, showing that the strategies had really been abstracted from their original settings.
- All tables had been explored after at most 11 minutes (scale and speed corresponding to the real settings).
- On average, 83% of the tables had been explored after 3 minutes.
- Dividing the ground in squares 20 pixels across[5], which corresponds to the average "width" of a human as seen on the videos, between 78% and 94% of all reachable squares had been reached at least once after 10 minutes, the actual average values varying according to maze size and complexity.
- These percentages increase with run duration.

Our goal was to show that the strategies of humans in a problem-solving situation are learnable, not to implement an efficient sweeper. We were pleased to see that the psychological bias "bottles are on tables and not on the ground" was learnt by our robot. We consider this a success even though it detracts from search efficiency when the treasure is on the ground.

For an example of possible practical application let us consider the vacuum-cleaner again. An elderly user always switches it off when the telephone rings, the better to hear. For the robot this is an unmistakable change of strategy because it goes from cleaning to idle. Automated feature extraction and selection would quickly learn the correlation between this change of strategy and its trigger, so once taught the robot would stop when the telephone rings and resume its work when the user puts the receiver down again.

7. CONCLUSION

The primary purpose of this work is showing that humans when they are in position of solving a problem make use of strategies that can be analysed and transferred to a robot. Obviously, it could be possible to reproduce the exact behaviour of the observed humans and we have the "robotic simulations" which do so, (noise excepted). These traces are very useful in order to compare what a "real human" does with what a "simulated human" does but they are useless to robotics since the humans are always observed in a particular setting and the slightest change in this setting would make the trace useless. Nor are they very useful to the psychologist since they are nothing more than a digitalised version of the video we started from. This is why it is necessary to analyse the human behaviour and to generalise it to problem solving strategies that can applied in any setting, and to other problems as well, as long as these strategies are meaningful for the new problem.

8. ACKNOWLEDGEMENTS

This work has been partially supported by the EU FP7 project HUMANOBS Grant number 231453.

9. REFERENCES

[1] A. Alissandrakis, C. L. Nehaniv, and K. Dautenhahn. Learning to imitate corresponding actions across dissimilar embodiments. *IEEE Transactions on Systems, Man, and Cybernetics*, 32:482–496, 2002.

[2] A. Alissandrakis, C. L. Nehaniv, and K. Dautenhahn. Correspondence mapping induced state and action metrics for robotic imitation. *Special Issue on Robot Learning by Observation, Demonstration and Imitation*, 2006.

[3] Billard and Hayes. Drama, a connectionist architecture for control and learning in autonomous robots. *Adaptive Behaviour*, 7:35–64, 1999.

[4] A. Billard and R. Siegwart. Robot learning from demonstration. *Robotics and Autonomous Systems*, 47:65–67, 2004.

[5] M. Brady. Artificial intelligence and robotics. *Artificial Intelligence Laboratory, MIT, A.I. Memo 756*, 1984.

[6] S. Calinon and A. Billard. *Imitation and Social Learning in robots, Humans and Animals: Behavioural, Social and Communicative Dimensions.* Cambridge University Press, 2007.

[7] S. Calinon, F. Guenter, and A. Billard. On learning, representing and generalising a task in a humanoid robot. *IEEE Transactions on Systems, Man and Cybernetics*, 37 (2), Part B, Special Issue on Robot Learning by Observation, Demonstration and Imitation, 2007.

[8] Demiris and Hayes. *Imitation in animals and artifacts.* 2001.

[9] R. Dillmann. Teaching and learning of robot tasks via observation of human performance. *Robotics and Autonomous Systems*, 47:109–116, 2004.

[10] V. Iemmi. Les apports des nouvelles technologies a la psychologie clinique: Les robots comme compagnons therapeutiques. 2005.

[11] H. Kraus. A formal theory of plan recognition and its implementation. *Reasoning About Plans, by J.F. Allen, H.A. Kautz, R.N. Pelavin, and J.D. Tennenberg*, pages 69–126, 1991.

[12] S. Schaal, A. Ijspeert, and A. Billard. Computational approaches to motor learning by imitation. *Philosophical Transactions of the Royal Society of London: series B, Biological Sciences*, 358:537–547, 2003.

[13] A. Sloman. Varieties of affect and the cogaff architecture schema. *Proceedings Symposium on Emotion, Cognition, and Affective Computing AISB'01 Convention*, pages 39–48, 2001.

[14] C. Tijus, N. Bredeche, Y. Kodratoff, M. Felkin, C. Hartland, V. Besson, and E. Zietti. Human heuristics for a team of mobile robots. *Proceedings of the 5th International Conference on Research, Innovation and Vision for the Future (RIVF'07)*, 2007.

[5] Due to lack of scaling we used pixels for our distance unit. 20 pixels roughly correspond to 30 cm.