The AAAI-17 Workshop on
Human-Aware Artificial Intelligence
WS-17-10

"Why Did You Do That?" Explainable Intelligent Robots

Raymond K. Sheh
Intelligent Robots Group, Department of Computing, Curtin University
Building 314, Kent St
Bentley WA 6102 Australia

Abstract

As autonomous intelligent systems become more widespread, society is beginning to ask: "What are the machines up to?". Various forms of artificial intelligence control our latest cars, load balance components of our power grids, dictate much of the movement in our stock markets and help doctors diagnose and treat our ailments. As they become increasingly able to learn and model more complex phenomena, so the ability of human users to understand the reasoning behind their decisions often decreases. It becomes very difficult to ensure that the robot will perform properly and that it is possible to correct errors.

In this paper, we outline a variety of techniques for generating the underlying knowledge required for explainable artificial intelligence, ranging from early work in expert systems through to systems based on Behavioural Cloning. These are techniques that may be used to build intelligent robots that explain their decisions and justify their actions. We will then illustrate how decision trees are particularly well suited to generating these kinds of explanations. We will also discuss how additional explanations can be obtained, beyond simply the structure of the tree, based on knowledge of how the training data was generated. Finally, we will illustrate these capabilities in the context of a robot learning to drive over rough terrain in both simulation and in reality.

Introduction

This paper presents preliminary work in developing an explainable artificial intelligent agent. This is an agent that is able to, at some level, explain its decisions and justify its actions to a human.

Much of the existing literature on explainable artificial intelligence tends to be concerned with higher level reasoning, data mining, diagnostics, natural language and argument construction or image recognition. In contrast, our work is concerned with low level behaviours and perception for robot control, and the generation of the underlying information that allows this to be explained. We refer to this as an explainable intelligent robot.

An intelligent robot that is explainable yields several important advantages.

Trust Humans tend to trust systems that they understand – or at least believe that they understand.

Accountability As systems become more mission-critical, society will increasingly want to know where the blame lies when things go wrong.

Ease of Debugging By making the decision processes transparent, it becomes easier for developers to discover and fix failures.

Training Explanations can help to teach users to perform a similar task or control a robot to achieve this task.

Discrepancy Resolution Related to training and debugging, where the user and agent differ in their decisions, the agent should be able to either convince the user that the agent is correct or give the user enough information to correct the error and understand its scope.

The application domain that we will use for this discussion is autonomous rough terrain mobility, which we evaluated both in simulation and on a real robot. This application is posed as a state-action machine learning problem. The state, or situation, consists of a combination of features drawn from past and present sensing as well as some encoding of the higher level goal and are presented as attributes to the machine learning algorithm. The output of the machine learning algorithm, be it a classification or a real value, represents the action that the robot should perform in order to best achieve the higher level goal, given its understanding of the current state.

We define rough terrain in terms of its effect on the robot's motion model. Rough terrain is terrain that affects the robot's motion to the point that to have any success, it must be explicitly accounted for. This is in contrast to terrain that can be classified into terrain that is undriveable and terrain that forms a drivable surface, perhaps with some simple modifiers such as maximum speed (Thrun et al. 2006).

The problem of rough terrain traversal is interesting from a machine learning perspective in general, and an explainable artificial intelligence perspective in particular, because it has several characteristics that challenge most traditional machine learning techniques. These include:

- There is often not a single correct action. Rather, actions range from being acceptable through to being undesirable. Therefore, very similar states may be labelled with different actions that are all correct. Thus crossvalidation tends to be a particularly poor indicator of real world per-

formance. Indications of expected performance therefore need to be more nuanced.

- Due to uncertainties in real-world sensing, features may be missing or have a low level of confidence. For example, features that correspond to the profile of the terrain in particular areas may be missing due to occlusions or a dirty sensor.
- A large number of features can make it very difficult to both learn and generate explanations. It can be impossible to distil the sensor data down into a small number of features that capture both the wide variety of situations as well as the nuances of each situation.

Our agent is structured as a Behavioural Cloning agent (Bratko, Urbančič, and Sammut 1998; Isaac and Sammut 2003; Kadous, Sammut, and Sheh 2006). The underlying model we use is based on decision trees. These are inherently explainable due to their natural ability to decompose their decisions into a sequence of IF-THEN-ELSE statements. Their expressive nature also allows them to leverage additional information from such sources as the training process itself in order to provide more intuitive and useful explanations.

Next we will present existing work on explainable artificial intelligence, especially as it relates to intelligent agents and robots. We will then explain the structure of our explainable intelligent robot and the capabilities that it offers. Finally we will present some preliminary examples of the explanations that can be obtained.

Background

Intelligent agents that can explain their actions and justify their decisions is fast becoming a popular topic for research (Defense Advanced Research Projects Agency (DARPA) 2016; Ribeiro, Singh, and Guestrin 2016; Johnson 1994). However, their roots date back to the early days of artificial intelligence research. In the 1980's, expert systems research was particularly focused on providing an explanation for decisions, with the ability to actually run the resulting models a secondary consideration.

Since then the focus of the artificial intelligence community has shifted towards models and techniques that emphasise predictive power while the ability to explain the decision processes has taken a back seat. In the limit, techniques such as K-nearest-neighbour do not seek to distill any meaning from the data while techniques such as those based on neural networks distribute any concept of meaning across data structures that are largely impenetrable to inspection.

In this work, we focus on the techniques to extract information from machine learned models for the purpose of generating explanations and justifications that help users to understand the decision making process. This is in contrast to other work on systems that try and persuade, negotiate, deliberate or quarrel with another (usually human) agent. (Parsons and McBurney 2003) presents additional discussion about the different ways in which explanations in inter-agent communication can be framed. In the referenced definitions by (Walton and Krabbe 1995) our work is perhaps best described by the category of "Information-Seeking Dialogues" where the human user has questions that the autonomous agent is believed to know the answers to.

Explaining more than a human can

An issue with early work in expert systems came from the problem that can be summarised as follows: "Experts don't know what they're talking about." In many applications, be it driving a car or making a medical diagnosis during a consultation, once the human has become an expert, the decision making processes will have moved into the subconscious mind.

Much of the workings of the subconscious mind are hidden from the conscious and thus unavailable to introspection. For example, as a human driving a car, you don't think "OK I will need to turn the steering wheel by 5° in the next 3 seconds", you just do it. Anything that a human can do without thinking and/or that requires practice for proficiency has a significant component of subconscious decision making.

Having the decision making processes hidden in the subconscious makes introspection as the basis for determining rules extremely difficult. This is especially the case where the decision making processes are based on inputs that are difficult to express the input in neat, symbolic terms, such as when driving a vehicle from a camera image. This is because the decomposition of the input into a relevant symbolic representation is also being done by the subconscious.

Behavioural Cloning

A field of machine learning, dubbed *Behavioural Cloning* (Bratko, Urbančič, and Sammut 1998; Isaac and Sammut 2003; Kadous, Sammut, and Sheh 2006), sought to address this problem. Behavioural Cloning is similar to techniques such as learning from demonstration (Atkeson and Schaal 2016) and apprentice learning (Abbeel and Ng 2004). These techniques take, as training data, state-action pairs, logged as a demonstrator performs the desired task. Behavioural Cloning has been applied to robot control problems that include controlling container cranes (Suc and Bratko 1999), flying planes (Isaac and Sammut 2003) and driving over rough terrain (Kadous, Sammut, and Sheh 2006).

They seek to learn some kind of model that can map from state to action. The goal is to predict, in some manner, what a demonstrator will do in a given situation. However, unlike other branches of learning from demonstration, the goal of Behavioural Cloning is to generate a model that can explain some aspect of the decision making process. The fact that the model can also be used to predict and, ultimately, replace, the expert is a useful byproduct of this process. The idea is that instead of asking an expert for the rules that make up an expert system, it may be more natural to have the expert demonstrate their skill and for the system to make up these rules.

Naturally, there are an infinite number of possible explanations that will explain any set of state-action pairs that a demonstrator can provide. In general Behavioural Cloning techniques trade off simplicity in the explanation for predictive accuracy. For example, decision trees and their variants

are a popular technique for generating the underlying models. This trade-off is encapsulated in the various pruning algorithms that simplify the resulting tree.

Modelling Non-Humans

Of course, the demonstrator in this case is purely a source of state-action pairs. Traditionally, this demonstrator has been a human, observing the situation through the same state attributes and selecting from the possible actions. However, there is nothing to say that this must be the case.

It is possible to use these techniques to attempt to explain the decision process of another autonomous policy. For example, in (Ribeiro, Singh, and Guestrin 2016) the behaviour of a black-box classifier is probed around a given decision point and the result used to create an explanation that is consistent with the local result. No attempt is made to fit the explanation more broadly across the black-box classifier's state space.

As we will see, it is also possible to model non-human demonstrators that are not physically realisable (Sheh 2010). For example, generating state-action pairs doesn't rely on the underlying system being causal. It is possible to generate state-action pairs using an A* search in a system, such as a simulator, that can be reset to an arbitrary state. Of course, at this point the cloned policy is not actually explaining the demonstrator at all. We will describe the meaning that can be ascribed to such a clone in further detail shortly.

Explainable Models

A prerequisite for an explainable artificial intelligence system is a model that is conducive to generating meaningful explanations. Of course, any model that can be executed by a computer can be explained to some degree. Even a neural network or K-nearest-neighbour based system can explain how it arrived at a given decision. The crucial point, however, is that explanations that focus on pure statistics, network weights and the like, don't provide us with information that increases our knowledge of the system. To paraphrase (Defense Advanced Research Projects Agency (DARPA) 2016), an explainable model should provide us with information that answers the following questions:

- Why did the agent do that and not something else?
- When does the agent succeed and when does it fail?
- When can I trust the agent?
- How do I correct an error in the agent?

In this work, we focus on the underlying machine learning techniques that can answer these questions. We use models that store most or all of their learned knowledge in structures that are conducive to being represented as logic rules. Of course, the following discussion also assumes that the features input to, and the classes or outputs expected of, these models are themselves conducive to human interpretation. They may be transparent transformations of underlying sensor data or intermediate representations that can still be understood by humans. Naturally, if this is not the case, then it is likely to be difficult to extract any meaning from the system as a whole.

There are of course other ways of framing this problem such as starting with the desired dialog itself and then working back to the representation required to support it. There is significant work in the natural language processing community concerning the development of arguments, justifications and other dialogs and then forming the information required to support this. For example, (Tolchinsky et al. 2012) presents a model that is focused on dialog in this manner. While there is the potential for future work in combining the two approaches, we consider this to be outside the scope of the current work, which will focus on ways of extracting additional useful information from the machine learned models, allowing the human user to ask the aforementioned questions and presenting the answers in a straightforward manner.

Decision Trees Decision trees (Quinlan 1993) are classifiers partition the state space by recursively splitting on one attribute at a time. Generally this is the attribute that, at the given point in the tree, best discriminates between two (or more) classes. Different decision tree learning techniques use different methods for determining when enough splits have been made and how to "prune" the tree back to avoid overfitting the training data. The mechanics behind how decision trees are learned lie beyond the scope of this paper and can be found in (Quinlan 1993).

In this paper, we will consider decision trees that also store, at each node, the class distribution observed during training. This information allows the tree to provide not just a classification of a novel example, but also a confidence value, derived by the distribution of the training examples observed at that leaf. It also allows the agent to highlight situations where several actions might well be acceptable, or where the agent is unable to distinguish between aliased states, as they will have uniformly high confidence values.

Decision trees are useful as components of models in explainable artificial intelligence systems because they are able to explain their decision processes in the form of IF-THEN-ELSE statements. Crucially, they also justify these decisions by way of compact information theoretic explanations based on the training data. As we will show, this is still largely possible even in a pruned tree.

An Explainable Intelligent Robot

The ultimate goal of our work in creating a framework for Explainable Intelligent Robots is to enable the intelligent agents that control these robots to carry on the kind of dialog that a human operator might carry out with an observer in explaining their decisions. For now, this will be limited to the types of questions cited earlier. The following is a hypothetical example of a dialog that might take place between a human and an explainable intelligent robot that is tasked with driving over rough terrain.

Robot: I have decided to turn left.

Human: Why did you do that?

Robot: I believe that the correct action is to turn left
BECAUSE:
I'm being asked to go forward

AND This area in front of me was 20 cm higher than me *highlights area*
AND the area to the left has maximum protrusions of less than 5 cm *highlights area*
AND I'm tilted to the right by more than 5 degrees.
Here is a display of the path through the tree that lead to this decision. *displays tree*

Human: How confident are you in this decision?

Robot: The distribution of actions that reached this leaf node is shown in this histogram. *displays histogram* This action is predicted to be correct 67% of the time.

Human: Where did the threshold for the area in front come from?

Robot: Here is the histogram of all training examples that reached this leaf. 80% of examples where this area was above 20 cm predicted the appropriate action to be "drive forward".

Human: Why didn't you consider the slope of the area to your left?

Robot: The slope of the area to the left was considered in these branches of the tree. *highlights decision nodes* For this area to be relevant, the minimal change would be a tilt to the right by less than 5 degrees OR the area to the left having maximum protrusions of more than 5 cm.

Human: Show me examples of situations, similar to the one you're in now, where the action to take would be to go forward.

Robot: The most similar path through the decision tree that will lead to "go forward" being the most likely action is this one. *highlights decision nodes* Examples that were observed at the leaf of this tree are as follows: ...

For now, these dialogs will be limited to queries expressed as a small number of phrases. These may be parameterised by selections on visualisations such as historic and synthesised examples of the feature space. Extending these dialogues to incorporate natural language processing is an important next step in improving the usability of such a system. At present we consider it outside the scope of this work, which focuses on the machine learning techniques and models that will provide the requisite information.

While we are not currently working on a true natural language system, it is important that these preset queries – and, thus, the information that the underlying system generates – reflect the types of questions that a human might ask another human. For example, a human is more likely to ask "Show me examples of situations that you regard as similar to what you see now" than " Please show me your network weights".

An important feature of this type of agent is that it not only presents to the user the specific decisions and attributes that were explicitly considered. This is due to the decision tree only considering one attribute at a time and ignoring all others. It also uses the training data that supported that decision in its justification. Not only does this mean that the user can be presented with concrete examples, it also helps the user to detect when unintended correlations have been used as part of the decision process. For example, it is possible to detect malfunctioning sensors or corrupted training data that might have become incorporated into the model as the resulting explanations will not make sense.

Attribute-centric explanations

Beyond providing a well defined framework for creating meaningful explanations based on the training data, decision trees are also able to highlight the attributes that tend to matter the most for different types of actions. The information theoretic nature of the tree generation process allows for the extraction of explanations that are attribute-centric rather than example-centric.

At its simplest, a decision tree is able to rank attributes according to overall importance. These are the attributes that appear most often and towards the top of the tree. Beyond this, by weighting these rankings according to the probability that they discriminate between a given action and others, it is also easy to determine attribute importance for a given action. A particular feature of decision trees is that they can perform a type of implicit dimensionality reduction, but do so in a way that is local to a particular area of the state space. This is in contrast to many other dimensionality reduction techniques, where attributes that are not generally useful are discarded even if they may be crucial in distinguishing between actions in a specific part of the state space.

This allows the user to engage the agent in a more nuanced dialog about the importance of different attributes. For example, the agent can answer the question "What are examples of states where this attribute becomes important?" or "When this attribute becomes important, what other attributes also matter?". Such probabilistic attribute-centric explanations further help to educate the user as to the types of information that the agent finds most useful.

Explaining the importance of missing attributes

It is not unusual for attributes to be missing in robotics applications. For example, in terrain traversal, attributes might be based on the shape of terrain in different areas around the robot. A particular area might not be sensed due to terrain self-occlusion, movement in the robot or even issues with the sensor such as dirt on a lens.

Decision trees can handle missing attributes more gracefully than most classifiers. During the course of traversing the tree, if the agent comes across a decision node on an attribute that is missing, the agent continues recursive traversal down both branches in parallel. When the recursion returns, the probability distribution over actions between the two branches are merged according to the probability distribution of the missing attribute seen at this node during training.

This process lends itself to novel ways of keeping users informed as to how degradation in the performance of its sensors affects its decision making abilities. For example, it is easy to present the user with metrics, derived from the distributions over actions before and after this merge, as a measure for how much additional confidence could be gained were the missing attribute present.

Cloning a Non-Causal Agent

A behavioural cloning agent is usually trained with examples of attributes that represent the state of the robot and its goals, and values that represent the action that the robot should take to follow the demonstrating agent's policy. A simulator, with the ability to save and re-load the entire world (including its random number generator) at arbitrary points, may be available. The simulated robot can be placed into similar starting states that the cloned agent will face. Simulated noisy sensor information can be generated, from which features may be extracted. A demonstration agent can then be developed that uses a search, such as an A* search, through possible actions in order to achieve the higher level goal. Such an agent will, by definition, be following the optimal policy with perfect information.

Such a demonstrator cannot be implemented in real life because the search process is non-causal and requires the simulated world to be reset to past states. At first glance it would also seem to violate an important property required in behavioural cloning, that of the demonstrator and cloned agent having access to the same information. In effect, the search based demonstrator has access to "perfect" information, including information that allows it to perfectly "predict" what will happen as a result of each of its actions.

This is problematic if we consider the machine learned policy to be an attempt at cloning the decision making processes of the demonstrator. However, we can instead consider the machine learned policy to be an attempt at approximating an optimal policy that has perfect information. Of course, there is a gap between the information that is available to the cloned agent and the demonstrator – the cloned agent has access to noisy, imperfect sensing and cannot predict the future in a stochastic environment. Instead, given sufficiently varied training data, the cloned policy predicts, given the noisy information that it does have, the distribution of correct actions to take, over the space of distinct situations and futures that it cannot observe.

We have now generated a cloned policy from nothing more than a simulator and the distribution of starting states. Explanations, of the type previously discussed, can still be extracted from such a policy. A particularly interesting interpretation of these explanations is that they represent meaning that the agent has discovered about the world and its task, through simulated experimentation. In effect, the agent has "imagined" or "envisaged" scenarios and run "thought experiments" to come up with an explanation of how the world works.

Such an approach opens up the possibility that this type of explainable artificial intelligence agent can learn about a problem, independent of any other knowledge on how to solve it, and come up with explanations that can teach us how to approach it.

Experiments

We have used the domain of autonomous rough terrain traversal to demonstrate these techniques. The real and simulated robots are shown in Figure 1. The performance of the simulator was validated according to the procedure described in (Pepper, Balakirsky, and Scrapper 2007). The high level goal was expressed in terms of a navigation corridor and a prescribed direction along that corridor, in global coordinates. Leaving the corridor, becoming stuck (as defined by moving less than a threshold amount down the corridor over a certain time) or flipping the robot were considered failures.

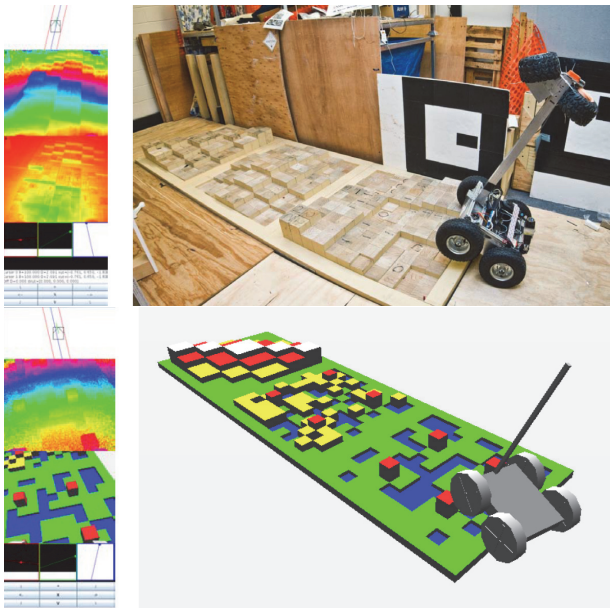

Figure 1: The real and simulated robots, environment and representation of the inputs into the agent.

The real and simulated robots were equipped with the following sensors:

2D Position Via a levelled Hokuyo URG-04LX laser rangefinder with scanmatching on the real robot and via simulator groundtruth with added noise in simulation.

3D Terrain Via a 3D range camera. CSEM SwissRanger SR3000 on the real robot, simulated range camera with added noise model in simulation. In both cases, 3D SLAM, assisted by the 2D position data, was used to preserve information about areas no longer visible.

Height Via 3D SLAM on the real robot and via simulator groundtruth with added noise in simulation.

Orientation Via an XSens IMU on the real robot and via simulator groundtruth with added noise in simulation.

From these sensors we extracted the following attributes.

Corridor Relative Yaw: The difference between the robot's heading and the direction of the navigation corridor.

Corridor Offset: The distance between the robot and the centreline of the navigation corridor, expressed as a percentage of corridor width, with sign indicating which side of the corridor the robot is on.

Pitch and Roll: Pitch and roll of the robot.

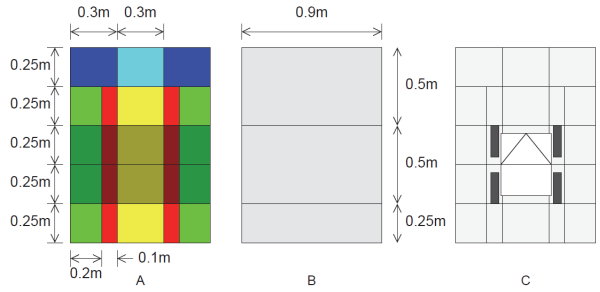

Figure 2: Regions of Interest for the terrain feature extractor (A, B). The robot, with forward facing up the page, is superimposed in (C).

Terrain features: The height, slope, direction of slope and maximum deviation from the plane for the terrain in 25 Regions of Interest around the robot as shown in Figure 2. Each Region of Interest must have at least a certain percentage of area observed to be valid, otherwise the feature is regarded as missing.

We have succeeded in generating cloned policies, using the decision tree learning techniques as described, using a human demonstrator in simulation, a human demonstrator on the real robot and the search based demonstrator in simulation. We used J48 as the decision tree learner, instrumented to provide the requisite information from which to extract the explanations. Full details of the process by which the model was learned, along with a full analysis of the overall performance of the cloned agent, appears in (Sheh 2010) and will not be repeated here. Instead, this paper focuses on the ability of the agent to extract meaningful explanations from the learned model and use it to answer the types of questions that a user might have.

The trees that were learned in practice run to several hundred nodes. For ease of explanation, we limit our preliminary discussion to more heavily pruned trees. Software to more easily present and analyse more extensive trees, and to present full statistics of the training data, is in development. Thus the extraction of explanations from the full tree is left for future work.

An example of such a heavily pruned tree, trained using the A* search based demonstrator, using a minimum of 30 examples per leaf and a pruning confidence of 0.001, is shown in Figure 3. Grey rectangles are leaf nodes. The first number represents the action that the robot should take (0 = drive forward and to the left, 1 = drive forward, 2 = drive forward and to the right, 3 = spin left, 4 = spin right, 5 = drive backwards and to the left, 6 = drive backwards, 7 = drive backwards and to the right). The second number represents the number of training examples reaching that leaf. The third represents the number of training examples that did **not** match this classification. Decision nodes that split on terrain features are displayed in cyan, those representing inertial (eg. pitch and roll) features in green and those representing the goal are in yellow. The high prevalence of goal-related nodes in this tree is a reflection of its unusually heavily pruned nature. In a full tree a significantly larger number of terrain features appear, albeit in the lower levels of the tree as they modify what would otherwise be a fairly simple corridor-centring algorithm.

With this tree, we can illustrate the kind of dialog presented earlier. For example:

Robot: I have decided to drive forward and to the right.

Human: Why did you do that?

Robot: I believe that the correct action is to drive forward and to the right BECAUSE:
I am headed between 7.5° and 18° relative to the corridor
AND between 19% and 75% to the left of the corridor
AND pitched up by more than 1.7°
AND the area in front of my left wheel is higher than normal ground level
AND the largest obstacle in that area is in the left 20%.

Human: How confident are you in this decision?

Robot: This action is predicted to be correct 61% of the time, based on 72 training examples and 8 possible actions.

Human: What is the next most likely action to take?

Robot: The next most likely action to take is to drive forward but that would require the largest obstacle in the area to the front of the left wheel to be in the right 80%.

By itself this dialog is already informative, especially to a human user who may not be familiar with reading IF-THEN-ELSE statements or the debugging visualisations provided by the decision tree learner. Future work will involve relating this to additional information as described previously and implementing the user interfaces that allow the agent to present the information in a more intuitive fashion such as by highlighting paths through the tree and areas in the sensor data.

Conclusion

We have presented an approach to building an explainable artificial intelligence agent for controlling a robot. Behavioural cloning couples a Learning from Demonstration framework with machine learning techniques such as decision trees, which create models that lend themselves to being explainable.

This technique has been applied to the problem of controlling a mobile robot to traverse rough terrain, based on onboard sensing and a higher level goal of a navigation corridor and direction. Robot control problems such as these pose unique machine learning challenges including missing attributes, situations where multiple actions may be acceptable, stochasticity and difficulty in generating training data. We have presented opportunities to generate relevant explanations that help users to understand these challenges.

We have also presented an alternative source of training data using a search based demonstrator. The explanations that are obtained from such a demonstrator have a unique property in that they represent knowledge that the agent has extracted from the simulator and distribution of situations presented to it. Such an approach opens up opportunities

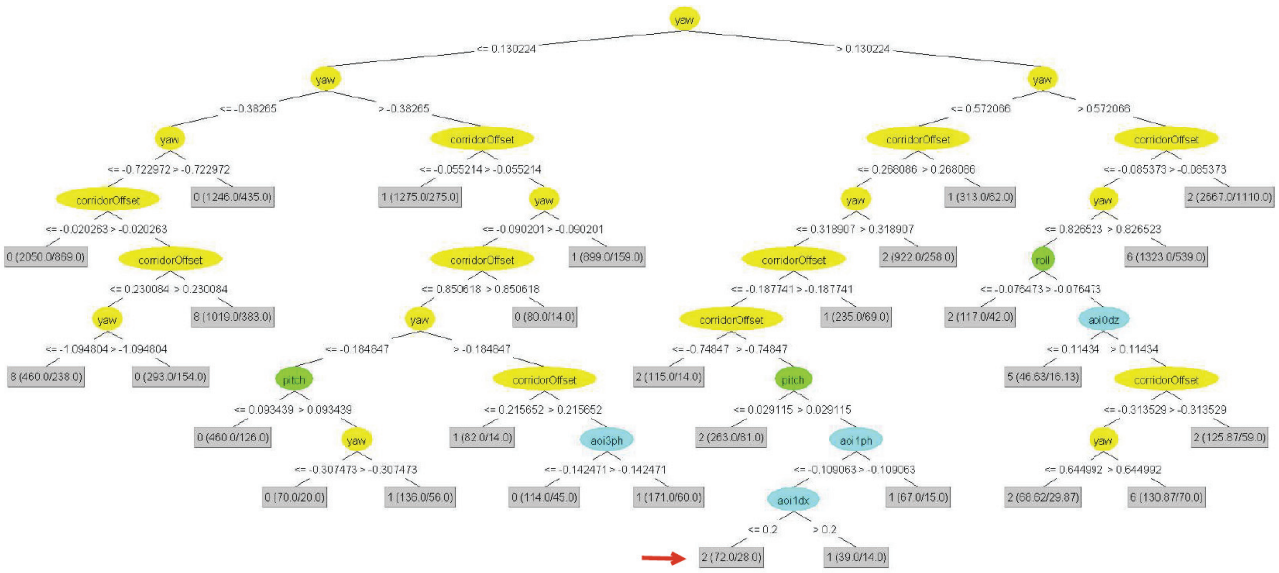

Figure 3: An example of a heavily pruned decision tree. The red arrow indicates the leaf node described in the example in the text.

for building explainable artificial intelligence agents that can teach themselves about problems that humans cannot currently solve, and then generate explanations that can teach us about the problem. Finally, although analysis of the overall learning technique is beyond the scope of this paper, we have presented an example of a preliminary explanation based on this application.

We are currently developing software that will allow more expressive explanations. This includes the presentation of additional statistics in the training data and attribute-centric explanations. Future work includes further analysis of these explanations, and user studies to determine their effectiveness in improving the acceptance and understanding of these systems.

References

Abbeel, P., and Ng, A. Y. 2004. Apprenticeship learning via inverse reinforcement learning. In *Proc. 21st Int. Conf. on Machine learning*. ACM.

Atkeson, C. G., and Schaal, S. 2016. Robot learning from demonstration. In *Proc. Int. Conf. on Machine Learning (ICML)*, volume 97.

Bratko, I.; Urbančič, T.; and Sammut, C. 1998. Behavioural cloning of control skill. *Machine Learning and Data Mining* 335–351.

Defense Advanced Research Projects Agency (DARPA). 2016. Broad Agency Announcement: Explainable Artificial Intelligence (XAI). online.

Isaac, A., and Sammut, C. 2003. Goal-directed learning to fly. In *Proc. Int'l. Conf. on Machine Learning*, 258–265.

Johnson, W. L. 1994. Agents that Learn to Explain Themselves. In *Proc. AAAI Conf. on Artificial Intelligence*, 1257–1263.

Kadous, M. W.; Sammut, C.; and Sheh, R. 2006. Autonomous traversal of rough terrain using behavioural cloning. In *Proc. 3rd Int. Conf. on Autonomous Robots and Agents*.

Parsons, S., and McBurney, P. 2003. Argumentation-based communication between agents. In *Communication in Multiagent Systems*, 164–178. Springer.

Pepper, C.; Balakirsky, S.; and Scrapper, C. 2007. Robot simulation physics validation. In *Proc. 2008 Workshop on Performance Metrics for Intelligent Systems*.

Quinlan, R. J. 1993. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *arXiv*.

Sheh, R. 2010. *Learning Robot Behaviours by Observing and Envisaging*. Ph.D. Dissertation, School of Computer Science and Engineering, The University of New South Wales, UNSW Sydney.

Suc, D., and Bratko, I. 1999. Modelling of control skill by qualitative constraints. In *Proc. 13th Int. Workshop on Qualitative Reasoning*, 212–220.

Thrun, S.; Montemerlo, M.; Dahlkamp, H.; Stavens, D.; Aron, A.; Diebel, J.; Fong, P.; Gale, J.; Halpenny, M.; Hoffmann, G.; et al. 2006. Stanley: The robot that won the darpa grand challenge. *Journal of field Robotics* 23(9):661–692.

Tolchinsky, P.; Modgil, S.; Atkinson, K.; McBurney, P.; and Cortes, U. 2012. Deliberation dialogues for reasoning about safety critical actions. In *Autonomous Agents and Multi-Agent Systems*, 209–259. Springer.

Walton, D., and Krabbe, E. C. W. 1995. *Commitment in Dialog*. SUNY Press.