Modelling accuracy and trustworthiness of explaining agents

Alberto Termine[1], Giuseppe Primiero[1], and Fabio Aurelio D'Asaro[1]

University of Milan, Milan, Italy
alberto.termine@unimi.it giuseppe.primiero@unimi.it
fabio.dasaro@unimi.it

Abstract. Current research in Explainable AI includes post-hoc explanation methods that focus on building transparent explaining agents able to emulate opaque ones. Such agents are naturally required to be accurate and trustworthy. However, what it means for an explaining agent to be accurate and trustworthy is far from being clear. We characterize accuracy and trustworthiness as measures of the distance between the formal properties of a given opaque system and those of its transparent explanantes. To this aim, we extend Probabilistic Computation Tree Logic with operators to specify degrees of accuracy and trustworthiness of explaining agents. We also provide a semantics for this logic, based on a multi-agent structure and relative model-checking algorithms. The paper concludes with a simple example of a possible application.

Keywords: Explainable AI · Accuracy · Trustworthiness.

1 Introduction

Within current research in AI, the alleged opacity of artificial systems based on machine learning methods (ML) has been widely discussed [5]. The inner structure of these systems is complex and providing useful explanations for them is a difficult task. Making ML systems more transparent, easier to survey and to check for correctness is the current aim of the field of *Explainable AI* (XAI).

In XAI, there is a common distinction between *opaque but comprehensible* systems and *full black box* systems [10]. The former include systems whose design is difficult to interpret, but implemented by algorithms whose inner structure is known. Among opaque but comprehensible systems we find many popular tools such as Bayesian nets and deep neural networks [2]. On the contrary, full black boxes are those systems implemented by algorithms whose inner strucuture is completely unknown or not a-priori available.

With respect to both levels of opacity, among the methods proposed in the literature to make ML systems more transparent, post-hoc explanations consist of building *ad-hoc* transparent explaining agents able to emulate, either locally or globally, the behaviour of a given opaque system. However, such agents are almost never able to provide exact explanations, because they usually require more computational resources than the opaque systems they explain [7]. Such

limitation is replaced by granting a certain degree of *accuracy*, i.e. conceding that the behaviour and the properties of the explaining agent should not be too far, according to some metric, from those of the system to be explained. Additionally, emulations conveyed by the explaining agents should be *trustworthy*, i.e. one should be able to explain the behaviour of explaining agents as well [5].

Despite the relevance of accuracy and trustworthiness to the task of offering useful post-hoc explanations, it is far from clear how to define, measure and evaluate these two properties. A common methodology relies on *fidelity*, i.e., measuring specific differences between the outputs of the explaining agent and of the system to be explained [7,5]. This is useful, albeit limited to explaining the observable behaviours of opaque systems and less appropriate to survey and check abstract properties. A different way to define, measure and evaluate degrees of accuracy and trustworthiness is therefore necessary for verification purposes.

In the present work, we propose alternative definitions of trustworthiness and accuracy, together with a framework to compute and approximate such measures. In particular, the behaviour of the explaining agents and the system to be explained is described in terms of discrete-time Markov chains [9,1] (DTMC, for short). At the same time, the properties to be verified are specified in the language of *Probabilistic Computation Tree Logic* (PCTL) [6] extended with weighted operators for accuracy and trustworthiness. Formulas of this new language called ATCTL (Computational Tree Logic for Accuracy and Trustworthiness) are evaluated in a multi-agent system semantics including a compact description of the behaviour of the system to be explained, which we call the *target-system*; as well as the behaviour of several explaining agents, which we call the *explanantes*, both in terms of DTMCs. In such a structure, the degree of accuracy for a given explaining agent against a certain property of the opaque system to be explained is defined as a measure of the distance between the set of states satisfying the property of interest reachable by either. In particular, we adopt the *Jaccard index* as our similarity measure. The degree of trustworthiness of an explaining agent against a certain property of the system to be explained, instead, is defined as a measure of the probability that eventually the system will reach a state satisfying the property of interest, provided that the explaining agent surely reaches such a state. Such probability is computed through the counting worlds technique, widely adopted in the model checking of multi-agent systems (see [3,8]). Such measures are easily extended to group operators for *common* and *distributed* degree of accuracy, respectively, trustworthiness, defined by analogy with the well-known notions of common and distributed knowledge in the logic of Interpreted Systems [3]. Finally, algorithms to model-check the degree of accuracy, respectively trustworthiness, of a given explaining agent, respectively a group of explaining agents, are introduced and evaluated through a small example of a possible application.

Notice that, to check degrees of accuracy and trustworthiness through the formalism here proposed requires that a compact description of the behaviour of both the opaque system to be explained and the explaining agents in terms

of a DTMC can be provided. Depending how such a compact description is obtained, we consider the problem of explanation with respect to different levels of opacity. A compact specification of the behaviour across time of these models in terms of DTMCs can be obtained by abstraction on the inner strucuture of the algorithm implementing them when this is known, i.e. for opaque but comprehensible models. Differently, for full black box models, an approximate matrix describing their behaviour across time can be inferred, in practice, through a sufficiently large number of observations of their input-output behaviour (see [4]).

The paper is structured as follows. In Section 2 we recall some background notions about Markov chains. In Section 3.1 we introduce the ATCTL language and present its syntax. In Section 3.2 we present the multi-agent structure over which we define the ATCTL semantics and introduce satisfiability conditions for formulas. In Section 4 we present efficient algorithms to check accuracy and trustworthiness. In Section 5 we demonstrate the checking algorithm on a small example. Finally, in Section 6, we point out some conclusive remarks about further developments of the formalism here introduced.

2 Background

In the following, we consider a multi-agent structure composed by agents that behave as *stochastic*, *time-homogeneous* and *memory-less* state-transition systems. To this aim we use discrete-time Markov chains (DTMC). In this section we recall some background notions about them, including a specific inference relation which is particularly relevant for model-checking purposes.

2.1 Markov Chains

Given a finite non-empty set of states $\mathcal{S}$, at each discrete time-step $t \in \mathbf{N}$, we describe an agent that shifts from a state $s \in \mathcal{S}$ to another, not necessarily different, state $s' \in \mathcal{S}$. The probabilistic transitions s, s' are *time-homogeneous*, meaning that the probability of a transition s, s' is independent from the time $t \in \mathbf{N}$ at which it occurs.

A path is a function $\pi : \mathbf{N} \mapsto \mathcal{S}$ whose values are the states reached by an agent at the various time-steps $t \in \mathbf{N}$. For simplicity, in the following we use π directly to denote the set of values of the function π, and we denote by $\pi(t)$ the state of the path π at time $t \in \mathbf{N}$. We collect all the possible paths π for an agent in a set Π that is endowed with a *σ-algebra* $\sigma(\Pi)$[1] and augmented to a probability space $(\Pi, \sigma(\Pi), P)$. Over this probability space we define a family $\{S_t\}_{t\in\mathbf{N}}$ of categorical stochastic variables such that $S_t : \pi \mapsto \pi(t)$ for each $t \in \mathbf{N}$. This family of variables describes the evolution across time of an agent. For each $S_t \in \{S_t\}_{t\in\mathbf{N}}$, $P(S_{t+1}|S_t)$ denotes a probability distribution

[1] In particular, $\sigma(\Pi)$ is usually the σ-algebra generated by the cylinder sets of Π that allows $\sigma(\Pi)$ to be always a measurable space (see [9,1]).

that assigns to each pair of states $s, s' \in \mathcal{S} \times \mathcal{S}$ the probability of an agent to reach state $s' \in \mathcal{S}$ at time $t+1 \in \mathbf{N}$ given that it is in state $s \in \mathcal{S}$ at time $t \in \mathbf{N}$. The time-homogeneous behaviour of the agent corresponds to assuming $P(S_{t+1}|S_t)$ to be the same for each $t \in \mathbf{N}$. Its memory-less condition corresponds to assuming the *Markov property*, i.e., $P(S_{t+1}|S_t, \ldots, S_0) = P(S_{t+1}|S_t)$.

Given the time-homogeneity and the Markov property, a compact specification of the stochastic behaviour of our agent across time can be achieved by means of an initial probability distribution $P(S_0)$ and a transition matrix $T : \mathcal{S} \times \mathcal{S} \mapsto [0, 1]$ whose elements are the values of $P(S_{t+1}|S_t)$ computed for each $s, s' \in \mathcal{S} \times \mathcal{S}$ and the choice of t is arbitrary because of time-homogeneity. The tuple DTMC:=$\langle \mathcal{S}, P(S_0), T \rangle$ composed by the set of states $\mathcal{S}$, the initial probability distribution $P(S_0)$ and the transition matrix T is called a *discrete-time Markov chain*. Since here we are interested in studying the properties of agents, we refer to so-called *labelled* DTMCs, that are obtained by extending standard ones with a finite non-empty set of labels AP used to represent elementary properties and a labelling function $l : \mathcal{S} \mapsto 2^{AP}$. From now on, when referring to DTMCs we always mean their labelled version.

2.2 Hitting Probability in Markov Chains

Given a DTMC:=$\langle \mathcal{S}, P(S_0), T \rangle$, a state $s \in \mathcal{S}$, and an event $A \subseteq \mathcal{S}$, the hitting probability of an event A with respect to s, denoted $h_A(s)$, is the probability of the agent described by the DTMC to reach eventually in the future at least one state $s' \in A$ from s. Given a finite time-horizon $t \in \mathbf{N}$, the bounded-time hitting probability of A is the probability of reaching *until* time-step $t \in \mathbf{N}$ at least one $s' \in A$ from s. It is possible to show that $h_A(s) = \lim_{t \to \infty} h_A^{\leq t}$. The existence of this limit is proved and its values correspond to the fixed point of $h_A^{\leq t}(s)$ (see [9]).

To efficiently compute $h_A^{\leq t}(s)$, let $\mathbf{T}$ denote the transition matrix obtained from T making all the states $s' \in A$ absorbing. A state $s \in \mathcal{S}$ is called *absorbing* if and only if $\forall s' \neq s, T(s, s') = 0$. Hence $h_A^{\leq t}(s)$ can be obtained as:

$$h_A^{\leq t}(s) := \sum_{s' \in A} \mathbf{T}^t(s, s') \tag{1}$$

The computational complexity of (1) is linear in t (see [1]). To compute the (unbounded) hitting probability $h_A(s)$, instead, we need to find the minimal solution to the following system of linear equations:

$$h_A(s) := \begin{cases} 1 \text{ if } s \in A\,, \\ \sum_{s' \in \mathcal{S}} T(s, s') \cdot h_A(s') \end{cases} \tag{2}$$

A minimal solution h_A of (2) satisfies the following conditions: (i) values of $h_A(s)$ for $s \in \mathcal{S}$ are a solution of the linear system (2), and (ii) any solution h'_A of the linear system (2) distinct from h_A satisfies $h'_A(s) \geq h_A(s)$ for all $s \in \mathcal{S}$. In practice, a minimal solution of (2) can be computed by solving a simple linear

programming task (see [1]) for each $s \in \mathcal{S}$. Since the complexity of each linear programming task to solve is linear with respect to $|\mathcal{S}|$, the overall computational complexity of computing a minimal solution of (2) is polynomial in $|\mathcal{S}|$.

3 ATCTL

3.1 Syntax

Definition 1 (ATCTL Syntax).

$$\phi := \top \mid p \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid P_{\nabla b}\psi$$

$$\psi := \bigcirc\phi \mid \phi_1 \bigcup \phi_2 \mid \phi_1 \overset{\leq t}{\bigcup} \phi_2$$

$$\epsilon := A^{e}_{\nabla h}\phi \mid CA^{\Gamma}_{\nabla h} \mid DA^{\Gamma}_{\nabla h} \mid T^{e}_{\nabla h}\phi \mid CT^{\Gamma}_{\nabla h}\phi \mid DT^{\Gamma}_{\nabla h}\phi$$

The ϕ and ψ fragments include usual PCTL formulae with their standard reading (see [1]). They specify characteristic properties of both the target-system M and the explanantes $e \in \mathbf{E}$, i.e., the agents emulating the behaviours of the target-system. We use ∇ as a meta-variable for $<, \leq, =, \geq, >$, b and h to denote real numbers in the interval $[0, 1]$. Notice that, the probabilistic operator $P_{\nabla b}$ expresses the weighted-probability of a path-formula ψ to hold quantified with respect to *all* the paths originating in a given state $s \in \mathcal{S}$ [2]

The ϵ fragment includes several weighted accuracy and trustworthiness operators either for an explanans $e \in \mathbf{E}$ or a group of explanantes $\Gamma \subseteq \mathbf{E}$. They express degrees of accuracy and trustworthiness of an explanans (resp., a group of explanantes) when they explain a certain property of the target-system, where this property is specified by the formula ϕ nested within the operators. The ϵ formulas have the following informal reading:

- $A^{e}_{\nabla h}\phi$: agent $e \in \Gamma$ has a degree of accuracy less/equal/greater than h in explaining ϕ;
- $CA^{\Gamma}_{\nabla h}\phi$: the group of agents Γ has a common degree of accuracy less/equal/greater than h in explaining ϕ;
- $DA^{\Gamma}_{\nabla h}\phi$: the group of agents Γ has a distributed degree of accuracy less/equal/greater than h in explaining ϕ
- $T^{e}_{\nabla h}\phi$: agent $e \in \Gamma$ has a degree of trustworthiness less/equal/greater than h in explaining ϕ;
- $CT^{\Gamma}_{\nabla h}\phi$: the group of agents Γ has a common degree of trustworthiness less/equal/greater than h in explaining ϕ;
- $DT^{\Gamma}_{\nabla h}\phi$: the group of agents Γ has a distributed degree of trustworthiness less/equal/greater than h in explaining ϕ.

[2] As in standard PCTL, the CTL existential and universal quantifiers, expressing quantification over paths satisfying a given formula ψ, here are omitted. It is easy to prove that they correspond to special cases of probabilistic quantification. In particular, $\exists\psi \iff P_{>0}\psi$ and $\forall\psi \iff P_{=1}\psi$. For the details, see [1].

3.2 ATCTL Semantics

In this section we present the ATCTL semantics, providing the multi-agent structure over which we define satisfiability conditions for the ATCTL formulas introduced above.

We consider a multi-agent structure including a finite set of agents $\mathcal{A} := \{M\} \cup \mathbf{E}$ where $M \notin \mathbf{E}$. We denote generic elements of $\mathcal{A}$ by i. The agents are abstract models of both the opaque target-system to be explained and the transparent systems used to explain it. In particular, we use M to denote the model of the target-system, while we use $\mathbf{E} := \{e_1, \ldots, e_n\}$ to denote a finite non-empty set of agents e, each one modelling a given explaining transparent system and that we call an *explanans*. With slight abuse of terminology, we will sometimes call the model M simply the target-system. We further model each agent $i \in \mathcal{A}$ as a DTMC $i := \langle \mathcal{S}, T^i, P^i(S_0), AP^i, l^i \rangle$. Note that all the agents share the same state space $\mathcal{S}$, while the transition matrix, the initial probability distribution, the set of labels and the labelling function are local to each agent $i \in \mathcal{A}$.

Definition 2 (Target-Explanans System (TES)). *The overall multi-agent system is a structure*

$$\mathcal{M}_{TES} := \langle \mathcal{S}, \mathcal{A}, \{T^i\}_{i\in\mathcal{A}}, \{P^i(S_0)\}_{i\in\mathcal{A}}, \bigcup_{i\in\mathcal{A}} AP^i, \{l^i\}_{i\in\mathcal{A}} \rangle$$

that includes a finite non-empty set of states $\mathcal{S}$, a finite non-empty set of agents $\mathcal{A}$, a family $\{T^i\}_{i\in\mathcal{A}}$ of transition matrices $T^i : \mathcal{S} \times \mathcal{S} \mapsto [0,1]$, one for each agent $i \in \mathcal{A}$, a family $\{P^i(S_0)\}_{i\in\mathcal{A}}$ of initial probability distributions $\mathcal{S} \mapsto [0,1]$, one for each agent $i \in \mathcal{A}$, a set of labels $\bigcup_{i\in\mathcal{A}} AP^i$ obtained as the union set of all the sets of labels AP^i of each agent $i \in \mathcal{A}$ and, finally, a family $\{l^i\}_{i\in\mathcal{A}}$ of labelling functions $l^i : \mathcal{S} \mapsto 2^{AP^i}$, one for each agent $i \in \mathcal{A}$.

Definition 3 (Satisfiability of ϕ and ψ formulae). *Given an agent $i \in \mathcal{A}$ and a state $s \in \mathcal{S}$ the following conditions hold:*

$i, s \models \top, \quad \forall s \in \mathcal{S}$
$i, s \models p$ iff $p \in l^i(s)$
$i, s \models \phi_1 \wedge \phi_2$ iff $i, s \models_i \phi_1$ and $i, s \models \phi_2$
$i, s \models \neg\phi$ iff $i, s \not\models \phi$
$i, \pi \models \bigcirc\phi$ iff $i, \pi(1) \models \phi$
$i, \pi \models \phi_1 \bigcup^{\leq t} \phi_2$ iff $\exists \tau \leq t : i, \pi(\tau) \models \phi_2$ and $\forall \tau' : 0 \leq \tau' < \tau$, $i, \pi(\tau') \models \phi_1$
$i, \pi \models \phi_1 \bigcup \phi_2$ iff $\exists \tau \geq 0 : i, \pi(\tau) \models \phi_2$ and $\forall \tau' : 0 \leq \tau' < \tau$, $i, \pi(\tau') \models \phi_1$
$i, s \models P_{\nabla b}\psi$ iff $P(i, s \models \psi)\nabla b$,

where $P(i, s \models \psi)$ denotes the probability that a path π originating in s (i.e., such that $\pi(0) = s$) satisfies ψ according to the agent $i \in \mathcal{A}$. Different methods to compute this probability are presented in Section 4.

We interpret the degree of accuracy as the *Jaccard index* computed over two specific set of states. Given two sets A and B, the *Jaccard index* of A and B, denoted by $J(A, B)$, is defined as:

$$J(A, B) := \frac{\mid A \cap B \mid}{\mid A \cup B \mid} \tag{3}$$

Notice that, whenever both $A = \emptyset$ and $B = \emptyset$, $J(A, B)$ is undefined. In such cases, we assume $J(A, B) = 1$. Given a TES $\mathcal{M}_{TES}$, we define $Sat_i(\phi)$ as the set of states $s' \in \mathcal{S}$ such that $i, s' \models \phi$. Given a state $s \in \mathcal{S}$, we denote by $Reach^i(s)$ the set of all the states $s' \in \mathcal{S}$ that are surely reachable from s for agent $i \in \mathcal{A}$. A state $s' \in \mathcal{S}$ is *surely reachable* from another state $s \in \mathcal{S}$ according to an agent $i \in \mathcal{A}$ if and only if

$$(h)^i_{\{s'\}}(s) = 1 \tag{4}$$

where $(h)^i_{\{s'\}}(s)$ is the hitting probability of the event $\{s'\}$ computed through the transition matrix T^i, i.e., the transition matrix describing the stochastic behaviour of the agent $i \in \mathcal{A}$, as by Equation (2).

To define the degree of accuracy with respect to a given property ϕ and a state $s \in \mathcal{S}$, we define

$$\Phi_i(s) := Sat_i(\phi) \cap Reach^i(s) \tag{5}$$

i.e., the set of states that satisfy property ϕ and are surely reachable from $s \in \mathcal{S}$, respectively, according to agent $i \in \mathcal{A}$. These sets generalize to a group of explanantes $\Gamma \subseteq \mathcal{A}$ as follows:

$$\Phi^C_\Gamma(s) := \bigcup_{e \in \Gamma} \Phi_e(s) \tag{6}$$

$$\Phi^D_\Gamma(s) := \bigcap_{e \in \Gamma} \Phi_e(s) \tag{7}$$

Definition 4 (Satisfiability of Accuracy formulae).

$$\mathcal{M}_{TES}, s \models A^e_{\nabla h}\phi \;\; \textit{iff} \;\; J(\Phi_M(s), \Phi_e(s)) \; \nabla h$$

$$\mathcal{M}_{TES}, s \models CA^\Gamma_{\nabla h}\phi \;\; \textit{iff} \;\; J(\Phi_M(s), \Phi^C_\Gamma(s))) \; \nabla h$$

$$\mathcal{M}_{TES}, s \models DA^\Gamma_{\nabla h}\phi \;\; \textit{iff} \;\; J(\Phi_M(s), \Phi^D_\Gamma(s))) \; \nabla h \, .$$

where $J(\Phi_M(s), \Phi_e(s))$ is the Jaccard index computed over the sets $\Phi_M(s)$ and, respectively, $\Phi_e(s)$, $\Phi^C_\Gamma(s)$ and $\Phi^D_\Gamma(s)$.

While accuracy is defined as a distance, trustworthiness is defined as a probability. In particular, the degree of trustworthiness of an explanans $e \in \mathbf{E}$ in explaining a property ϕ of a target-system M is defined as the probability that eventually the target system will reach a state satisfying ϕ, provided that the explaining agent surely reaches such a state. Given a set of explanantes $\Gamma \subseteq \mathbf{E}$, analogous definitions are provided for common and distributed trustworthiness referring to the *union*, respectively the *intersection set*, of all explanantes $e \in \Gamma$.

Definition 5 (Satisfiability of Trustworthiness formulae).

$$\mathcal{M}_{TES}, s \models T^{e}_{\nabla h}\phi \quad \textit{iff} \quad P(\Phi_M(s) \mid \Phi_e(s))\nabla h$$

$$\mathcal{M}_{TES}, s \models CT^{\Gamma}_{\nabla h} \quad \textit{iff} \quad P(\Phi_M(s) \mid \bigcup_{e\in\Gamma} \Phi_e(s))\nabla h$$

$$\mathcal{M}_{TES}, s \models DT^{\Gamma}_{\nabla h} \quad \textit{iff} \quad P(\Phi_M(s) \mid \bigcap_{e\in\Gamma} \Phi_e(s))\nabla h$$

where: $P(\Phi_M(s) \mid \Phi_e(s))$ is the probability of the event $\Phi_M(s)$ given the event $\Phi_e(s)$. We compute this probability through the well-known *counting-worlds* technique (see [3]) and by adopting a classical interpretation of probability, as follows:

$$P(\Phi_M(s) \mid \Phi_e(s)) := \frac{\mid \Phi_M(s) \cap \Phi_e(s) \mid}{\mid \Phi_e(s) \mid}. \tag{8}$$

Respectively for groups of agents:

$$P(\Phi_M(s) \mid \bigcup_{e\in\Gamma} \Phi_e(s)) := \frac{\mid \Phi_M(s) \cap \bigcup_{e\in\Gamma} \Phi_e(s) \mid}{\mid \bigcup_{e\in\Gamma} \Phi_e(s) \mid}, \tag{9}$$

$$P(\Phi_M(s) \mid \bigcup_{e\in\Gamma} \Phi_e(s)) := \frac{\mid \Phi_M(s) \cap \bigcap_{e\in\Gamma} \Phi_e(s) \mid}{\mid \bigcap_{e\in\Gamma} \Phi_e(s) \mid}. \tag{10}$$

Notice that, whenever $\Phi_e(s)$, $\bigcup_{e\in\Gamma} \Phi_e(s)$ or $\bigcap_{e\in\Gamma} \Phi_e(s)$ are the empty-set the above probability is undefined. In such cases, we assume it to be equal to 0.

4 Model-Checking

In this section we describe feasible procedures to model-check a given $\mathcal{M}_{TES}$ against properties specified in the ATCTL language. Let $\Lambda := \phi \mid \epsilon$ denote a generic ATCTL state formula. Given a TES $\mathcal{M}_{TES}$, a formula Λ and a state $s \in \mathcal{S}$ our task is to check whether s satisfies Λ. Let $Sat(\Lambda)$ denotes the set of all the states $s \in \mathcal{S}$ such that $\mathcal{M}_{TES}, s \models \Lambda$ and let λ denotes a generic sub-formula of Λ. The main algorithm works as follows[3]:

1. Generate the *parse tree* of Λ, decomposing Λ in its sub-formulas λ.
2. Traverse the parse tree of Λ visiting all the sub-formulas λ, starting from the leaves and working backwards to the roots,
3. At each sub-formula λ, calculate $Sat(\lambda)$,
4. Calculate $Sat(\Lambda)$ by composition of the various $Sat(\lambda)$,
5. Check whether $s \in Sat(\Lambda)$.

To this aim, we define a procedure to compute $Sat(\lambda)$ for all kinds of ATCTL state formulas (i.e., ϕ and ϵ).

[3] Notice that path-formulas ψ are usually not considered in a typical probabilistic model-checking workflow. For the details of the procedure, see [1].

4.1 ϕ fragment

When $\lambda := \phi$, we compute $Sat_i(\lambda)$ by an iterative application of the following recursive schema:

Definition 6 (Sat_i).

$$\begin{aligned}
&Sat_i(\top) := \mathcal{S}\\
&Sat_i(p) := \{s \in \mathcal{S} : p \in l^i(s)\}\\
&Sat_i(\phi_1 \wedge \phi_2) := Sat_i(\phi_1) \cap Sat_i(\phi_2)\\
&Sat_i(\neg\phi) := \mathcal{S} \setminus Sat(\phi)\\
&Sat_i(P_{\nabla b}\psi) := \{s \in \mathcal{S} : P(i, s \models \psi)\nabla b\}
\end{aligned}$$

The only non-trivial step is the computation of $P(i, s \models \psi)$. The procedure to compute $P(i, s \models \psi)$ varies depending on ψ:

- When $\psi := \bigcirc\phi$, this probability is computed as

$$P(i, s \models \bigcirc\phi) := \sum_{s' \in Sat_i(\phi)} T^i(s, s') \tag{11}$$

- When $\psi := \phi_1 \bigcup^{\leq t} \phi_2$, this probability corresponds to the the bounded-time hitting probability $(\mathbf{h}^i)^{\leq t}_{Sat(\phi_2)}(s)$ computed through a modified transition matrix $\mathbf{T}^i$ that is obtained from T^i making all the states $s' \in \mathcal{S}$ absorbing, excluding those in $Sat_i(\phi_1) \setminus Sat_i(\phi_2)$. The algorithm computes $(\mathbf{h}^i)^{\leq t}_{Sat_i(\phi_2)}$ by generating the modified transition matrix $\mathbf{T}^i$ and then computing

$$(\mathbf{h}^i)^{\leq t}_{Sat_i(\phi_2)}(s) := \sum_{s' \in Sat_i(\phi_2)} (\mathbf{T}^i)^t(s, s') \tag{12}$$

- When $\psi := \phi_1 \bigcup \phi_2$, this probability corresponds to $h^i_{Sat_i(\phi_2)|Sat_i(\phi_1)}(s)$ [4], i.e. the (unbounded) hitting probability of the event $Sat_i(\phi_2)$ with the additional condition that all the states visited before reaching an $s' \in Sat_i(\phi_2)$ are in $Sat_i(\phi_1)$. This can be obtained by computing the minimal[5] solutions of the following system of linear equations:

$$h^i_{Sat_i(\phi_2)|Sat_i(\phi_1)}(s) := \begin{cases} 1 \;\text{ if }\; s \in Sat_i(\phi_2)\,, \\ 0 \;\text{ if }\; s \notin Sat_i(\phi_1)\,, \\ \sum_{s' \in Sat_i(\phi_1)} T(s, s') \cdot h^i_{Sat_i(\phi_2)|Sat_i(\phi_1)}(s') \;\text{ otherwise}\,. \end{cases} \tag{13}$$

In practice, it is possible to compute the minimal solutions of the above system by solving a simple optimization task for each $s \in \mathcal{S}$ through linear programming (see [1]).

[4] Notice that, this must not be intended as a conditional probability.
[5] Here, minimality is defined as for Equation (2).

4.2 ϵ fragment

In this section, we define and comment an algorithm to compute $Sat(\epsilon)$.

Algorithm 1 takes in input a TES $\mathcal{M}_{TES}$ and a formula ϵ (line 1). Given an $s \in \mathcal{S}$, it computes $\Phi_M(s)$ by checking for each $s' \in Sat_M(\phi)$ whether $h^M_{\{s'\}}(s) = 1$, i.e., whether s' is reachable from s according to the target-system M (line 4). Notice that, computing $h^M_{\{s'\}}(s)$ through Equation (2) at line 5 requires an amount of time polynomial in $|\mathcal{S}|$ for each $s' \in Sat_M(\phi)$ (see Section 2). Since $Sat_M(s) \subseteq \mathcal{S}$, the time complexity of the overall procedure described in line 4 is polynomial in $|\mathcal{S}|$. Then the algorithm for each $e \in \Gamma$ computes $\Phi_e(s)$ by checking for each $s' \in Sat_e(\phi)$ whether $h^e_{\{s'\}}(s)$ again through the procedure described in Equation (2) (lines 9-11). The time complexity of computing $h^e_{\{s'\}}(s)$ for each $s' \in Sat_e(\phi)$ is therefore polynomial in $|\mathcal{S}|$. Since the procedure has to be iterated for each $e \in \Gamma$, the final time complexity of the overall procedure is polynomial in $|\mathcal{S}|$ and linear in $|\Gamma|$. Then the algorithm checks whether $s \models \epsilon$ (line 17) as follows: (i) switch on the proper satisfiability condition depending on the nature of ϵ, (ii) perform a simple sequence of algebraic operations on sets and, (iii) check whether the obtained result respects the threshold ∇h specified in the formula. Since this procedure consists of a very small number of simple algebraic operations on sets, it does not increase the time complexity of the overall procedure that remains polynomial in $|\mathcal{S}|$ and linear in $|\Gamma|$. Finally, to compute $Sat(\epsilon)$ the above described procedure has to be iterated for each $s \in \mathcal{S}$. Consequently, the overall time complexity of computing $Sat(\epsilon)$ is polynomial in $|\mathcal{S}|$ and polynomial in $|\Gamma|$.

5 Example

Let us consider as target-system a *probabilistic decision-tree* classifier [6] whose task is to predict whether a given patient might develop schizophrenia based on the following Boolean parameters: *gender*, *genetic disposition*, and *presence of correlated psychiatric disorders*. We use labels m for "male", g for "presence of genetic disposition", d for "presence of psychiatric disorders" and p for "being schizophrenic".

The behaviour of the classifier can be described by a DTMC M defined over $\mathcal{S}$ and provided with: (i) a set of labels $AP^M := \{m, g, d, p\}$, (ii) a labelling function l^M and, (iii) a transition matrix T^M described in Figure 1. There are sixteen different reachable states, i.e., $\mathcal{S} := \{s_0, s_1, \ldots, s_{15}\}$ stating the different profiles of the patients predictable given the analysis of the above parameters. We further assume that, according to the labelling l^M the only state that satisfies property m is s_{12}. Let P be a patient whose actual profile is $\langle m, g, d, \neg(p)\rangle$ that corresponds to the labels of $s_0 \in \mathcal{S}$, i.e., $l^M(s_0) = \{m, g, d, \neg(p)\}$. We see from the matrix T^M that, whenever the model receives the input $\{m, g, d, \neg(p)\}$ it returns

[6] This represents a typical example of a stochastic machine learning model. According to the classification we propose in the introduction, it can be classified as an *opaque but comprehensible* model. For more details, see [2].

Algorithm 1: $Sat(\epsilon)$

Input: $\mathcal{M}_{TES}$, ϵ
Output: $Sat(\epsilon)$

$Sat(\epsilon) \leftarrow \{\}$;
foreach $s \in \mathcal{S}$ **do**
  $\Phi_M(s) \leftarrow \{\}$;
  foreach $s' \in Sat_M(\phi)$ **do**
    if $h^M_{\{s'\}}(s) = 1$ **then**
      $\Phi_M(s) \leftarrow \Phi_M(s) \cup \{s'\}$
    end
  end
  foreach $e \in \Gamma$ **do**
    $\Phi_e(s) \leftarrow \{\}$;
    foreach $s' \in Sat_e(\phi)$ **do**
      if $h^e_{\{s'\}}(s) = 1$ **then**
        $\Phi_e(s) \leftarrow \Phi_e(s) \cup \{s'\}$
      end
    end
  end
  switch ϵ **do**
    case $A^e_{\nabla h}$ **do**
      if $J(\Phi_M(s), \Phi_e(s))\nabla h$ **then**
        $Sat(\epsilon) \leftarrow Sat(\epsilon) \cup \{s\}$
      end
    end
    case $CA^\Gamma_{\nabla h}$ **do**
      if $J(\Phi_M(s), \bigcup_{e\in\Gamma} \Phi_e(s))\nabla h$ **then**
        $Sat(\epsilon) \leftarrow Sat(\epsilon) \cup \{s\}$
      end
    end
    case $DA^\Gamma_{\nabla h}$ **do**
      if $J(\Phi_M(s), \bigcap_{e\in\Gamma} \Phi_e(s))\nabla h$ **then**
        $Sat(\epsilon) \leftarrow Sat(\epsilon) \cup \{s\}$
      end
    end
    case $T^e_{\nabla h}$ **do**
      if $P(\Phi_M(s) \mid \Phi_e(s))\nabla h$ **then**
        $Sat(\epsilon) \leftarrow Sat(\epsilon) \cup \{s\}$
      end
    end
    case $CT^\Gamma_{\nabla h}$ **do**
      if $P(\Phi_M(s) \mid \bigcup_{e\in\Gamma} \Phi_e(s))\nabla h$ **then**
        $Sat(\epsilon) \leftarrow Sat(\epsilon) \cup \{s\}$
      end
    end
    case $DT^\Gamma_{\nabla h}$ **do**
      if $P(\Phi_M(s) \mid \bigcap_{e\in\Gamma} \Phi_e(s))\nabla h$ **then**
        $Sat(\epsilon) \leftarrow Sat(\epsilon) \cup \{s\}$
      end
    end
  end
end
return $Sat(\epsilon)$

Table 1. T^M

0.25	0.25	0.25	0.25	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0.5	0.5	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0.25	0	0	0.25	0.25	0.25	0	0	0	0	0	0	0
0	0	0	0	0.5	0	0	0	0	0.5	0	0	0	0	0	0
0	0	0	0	0	0.5	0	0	0	0	0.5	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0.5	0.5	0	0	0
0	0	0	0	0	0	0	0.5	0	0	0	0	0	0.5	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0.5
0	0	0	0	0	0	0	0	0	0.5	0.5	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0.5	0.5	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0.5	0.5	0	0	0
0	0	0	0	0	0	0.5	0	0	0	0	0	0.5	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0.5	0.5	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0.5	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0.5

the output p. In our framework, this corresponds to saying that the model M and the state $s_0 \in \mathcal{S}$ satify the property $\phi := P_{=1}\top \bigcup(p)$. Now, suppose that one is interested in explaining the behaviour of M on input $\{m, g, d, \neg(p)\}$, i.e., in understanding why $M, s_0 \models \phi$. To this aim, she builds an explanans e (e.g. a rule-based system) [7] able to emulate the behaviour of the classifier locally on s_0. The behaviour of e can be modelled by another DTMC defined over $\mathcal{S}$, provided with the same set of labels and the same labelling function of M but described by a different transition matrix T^e that is reported in Figure 2. We are interested in evaluating whether the explanation of M's behaviour against property ϕ is more accurate than a desirable threshold, for instance ≥ 0.75. First, we build a TES $\mathcal{M}_{TES}$ that includes the model M of the classifier and the model of the explanans e. Hence, we check whether $\mathcal{M}_{TES}$ and $s_0 \in \mathcal{S}$ satify the following formula:

$$\epsilon := A^e_{\geq 0.75} P_{=1}\top \bigcup(p) \tag{14}$$

expressing our desirable requirement on accuracy. To evaluate whether $\mathcal{M}_{TES}, s_0 \models \epsilon$, we compute $Sat(\epsilon)$ through algorithm 1 and check whether $s_0 \in Sat(\epsilon)$. Since this is the case, we can conclude that e is sufficiently accurate.[8]

6 Conclusion

In this paper we presented a framework to model and check accuracy and trustworthiness of explaining agents. Among further possible developments we mention: the extension of the language with operators to specify degrees of transparency of systems with respect to their stakeholders and extensions of the semantics to model systems whose behaviour cannot be described through DTMCs

[7] Remeber that an explanans is an agent able to (locally) emulate the behaviour of the target-system and usually consider more transparent than this one.

[8] A Python implementation of Algorithm 1 is available at `https://github.com/dasaro/ATCTL` together with details on how to reproduce the results from the example.

Table 2. T^e

0.25	0.25	0.25	0.25	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0.5	0.5	0	0	0	0	0	0	0	0	0	0
0	0	0.5	0	0	0	0.5	0	0	0	0	0	0	0	0	0
0	0	0	0.25	0	0	0.25	0.25	0.25	0	0	0	0	0	0	0
0	0	0	0	0.5	0	0	0	0	0.5	0	0	0	0	0	0
0	0	0	0	0	0.5	0	0	0	0	0.5	0	0	0	0	0
0	0	0.5	0	0	0	0	0	0	0	0	0.25	0.25	0	0	0
0	0	0	0	0	0	0	0.5	0	0	0	0	0	0.5	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0.5
0	0	0	0	0	0	0	0	0	0.5	0.5	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0.5	0.5	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0.5	0.5	0	0	0
0	0	0	0	0	0	0.5	0	0	0	0	0	0.5	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0.5	0.5	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0.5	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0.5

but requires, for instance, continuous-time Markov chains, Markov decision processes or hidden Markov models.

Acknowledgments. This research has been funded by the Department of Philosophy "Piero Martinetti" of the University of Milan under the Project "Departments of Excellence 2018-2022" awarded by the Ministry of Education, University and Research (MIUR). The authors also thankfully acknowledge the support of the Italian Ministry of University and Research (PRIN 2017 project n. 20173YP4N3).

References

1. Baier, C., Katoen, J.: Principles of model checking. MIT Press (2008)
2. Bishop, C.M.: Pattern recognition and machine learning, 5th Edition. Information science and statistics, Springer (2007), https://www.worldcat.org/oclc/71008143
3. Chen, T., Primiero, G., Raimondi, F., Rungta, N.: A computationally grounded, weighted doxastic logic. Studia Logica **104**(4), 679–703 (2016)
4. D'Asaro, F.A., Primiero, G.: Probabilistic typed natural deduction for trustworthy computations. In: Proceedings of the 22nd International Workshop on Trust in Agent Societies (TRUST2021 @ AAMAS) (2021)
5. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. ACM Comput. Surv. **51**(5), 93:1–93:42 (2019). https://doi.org/10.1145/3236009, https://doi.org/10.1145/3236009
6. Hansson, H., Jonsson, B.: A logic for reasoning about time and reliability. Formal Aspects of Computing **6**(5), 512–535 (1994)
7. Linardatos, P., Papastefanopoulos, V., Kotsiantis, S.: Explainable AI: A review of machine learning interpretability methods. Entropy **23**(1), 18 (2021). https://doi.org/10.3390/e23010018, https://doi.org/10.3390/e23010018
8. Lomuscio, A., Raimondi, F.: MCMAS: A model checker for multi-agent systems. In: Hermanns, H., Palsberg, J. (eds.) Tools and Algorithms for the Construction and Analysis of Systems, 12th International Conference, TACAS 2006 Held as Part

of the Joint European Conferences on Theory and Practice of Software, ETAPS 2006, Vienna, Austria, March 25 - April 2, 2006, Proceedings. Lecture Notes in Computer Science, vol. 3920, pp. 450–454. Springer (2006)
9. Revuz, D.: Markov chains. Elsevier (2008)
10. Rudin, C., Chen, C., Chen, Z., Huang, H., Semenova, L., Zhong, C.: Interpretable machine learning: Fundamental principles and 10 grand challenges. CoRR **abs/2103.11251** (2021), https://arxiv.org/abs/2103.11251