Artificial Intelligence and Law manuscript No.
(will be inserted by the editor)

Identifying prohibition norms in agent societies

Authors Anonymous

Received: date / Accepted: date

Abstract In normative multi-agent systems, the question of "how an agent identifies norms in an open agent society" has not received much attention. This paper aims at addressing this question. To this end, this paper proposes an architecture for norm identification for an agent. The architecture is based on observation of interactions between agents. This architecture enables an autonomous agent to identify prohibition norms in a society using the Prohibition Norm Identification (PNI) algorithm. The PNI algorithm uses association rule mining, a data mining approach to identify sequences of events as candidate norms. When a norm changes, an agent using our architecture will be able to modify the norm and also remove a norm if it does not hold in the society. Using simulations of a park scenario we demonstrate how an agent makes use of the norm identification framework to identify prohibition norms[1].

Keywords norms · agents · architecture · norm identification · prohibition norms · simulation · societies · Normative Multi-Agent Systems (NorMAS)

1 Introduction

No other concept is invoked by social scientists more frequently than that of "norms" [70]. In human societies, norms have played an important role in regulating the behaviour of the individuals in a society. For example, regulation of behaviour in public spaces such as parks and restaurants is governed by norms. It is expected that no one litters a park. In a restaurant it is expected that no one smokes[2] and also one might be obliged to tip a waiter for a good service. Thus, norms are the societal rules that govern the prescription and proscription of certain behaviour. Norms also facilitate cooperation [33] and coordination [11] in human societies. Norms are fundamental

Address(es) of author(s) should be given

[1] We note that an early draft of this paper is available online as a discussion paper (which is not a formal publication). A related but distinct contribution on identifying obligation norms appears elsewhere (not cited to preserve anonymity).

[2] This is true in most countries.

to the understanding of social control and its variation across societies [19]. Hence, it is not surprising that norms are of interest to researchers in different disciplines such as sociology, economics, anthropology and computer science.

Inspired by human societies, researchers in multi-agent systems have investigated how the concept of norms can be used to constrain the behaviour of *software agents* [47] and also improve cooperation and coordination among agents [68]. Norms are simple constructs that have also been shown to reduce the amount of computation required by software agents, since, on identifying norms agents can follow norms without much thought [34].

Researchers in multi-agent systems are interested in norms from two perspectives, the top-down and bottom-up approaches. In the top-down approach to norms, researchers have investigated how normative architectures and systems can be constructed where norms are viewed mostly as constraints on the actions that an agent can perform. Constraining the actions that agents can perform can promote a smoother functioning society (e.g. constraining agents from littering a park). In the bottom-up approach to study of norms, researchers are interested in studying how norms may spread and emerge in agent societies based on their interactions with other agents in the society.

With the advent of Internet, software agents exist in electronic societies that are open and dynamic. Currently, multi-agent system researchers investigate how norm-based mechanisms can be used to facilitate social control in these electronic societies. Examples of norm-governed electronic societies include electronic institutions [6], virtual societies such as Second Life [28] and massively multi-player online games (MMOGs) [44]. Researchers are also investigating mechanisms for norm emergence in these societies [62, 67].

A limitation of the existing works on norms is that most works have concentrated on how norms regulate behaviour [36, 47]. These works assume that agents somehow know (a priori) what the norms of a society are (e.g. through off-line design). For example, an agent may have obtained the norm from a leader [16] or through an institution that prescribes what the norms of the society should be [3, 24, 37, 69, 75]. However, it may not be possible for an agent designer to specify all possible norms that agents may encounter in an open agent society. Norms in societies may emerge dynamically. Additionally, as the composition of an open agent society changes the norms may change because of agents joining and leaving the society. Hence, there is a need for an explicit norm identification mechanism for an individual agent to recognize or identify norms in these open agent societies. The agents in these societies should have the capability to recognize a norm when it is established and also recognize when it changes. The agents should also be able to add, modify and remove norms dynamically. To that end, this paper discusses the internal agent architecture for norm identification. An agent using this architecture uses an association mining approach to identify prohibition norms.

It should be noted that, our investigation in this work is on how *artificial software agents* that are situated in a digital environment learn the norms of the society[3]. The

[3] In this work, we do not model how humans learn about norms in societies. However, the model proposed for artificial agents is inspired by how humans learn (e.g. observational learning, experiential learning and communication-based learning [41].)

norm identification mechanism will enable an agent to operate well in the community it is situated in (i.e. upon learning about a norm, it may decide to abide by the norm thus facilitating smoother functioning of the society and also avoiding penalties for not following the norm).

The paper is organized as follows. Section 2 provides a background on norms in human societies and how the concept of norms is investigated in the field of normative multi-agent systems (NorMAS). It also discusses the related work. Motivational scenarios for the work on norm identification are presented in Section 3. Section 4 provides an overview of the process associated with identifying norms and Section 5 describes the components of the norm identification framework. Section 6 explains the attributes of the norm identification framework. A detailed discussion on the two main components of the norm identification framework namely candidate norm identification and norm verification are discussed in Sections 7 and 8, respectively. Section 9 discusses the experimental results on norm identification. Section 10 provides a discussion on the work that has been achieved and the issues that need to be addressed in the future. Concluding remarks are presented in section 11.

2 Background and related work

This section provides a background on norms in human societies and multi-agent systems. The works related to our work on norm identification are also discussed.

2.1 Background on norms in human societies

Due to the multi-disciplinary nature of norms, several definitions for norms exist. Habermas [40] identified norm-regulated actions as one of the four action patterns in human behaviour. A norm to him means *fulfilling a generalised expectation of behaviour*, which is a widely accepted definition for social norms. A behavioural expectation is *generalized* if every member of a social group expects all others to behave in a certain way in a given situation. Ullmann-Margalit [74] describes a social norm as a prescribed guide for conduct or action which is generally complied with by the members of the society. She states that norms are the result of complex patterns of behaviour of a large number of people over a protracted period of time. Coleman [23] writes *"I will say that a norm concerning a specific action exists when the socially defined right to control the action is held not by the actor but by others"*. Elster notes the following about social norms [33]: *"For norms to be social, they must be shared by other people and partly sustained by their approval and disapproval. They are sustained by the feelings of embarrassment, anxiety, guilt and shame that a person suffers at the prospect of violating them. A person obeying a norm may also be propelled by positive emotions like anger and indignation ... social norms have a grip on the mind that is due to the strong emotions they can trigger"*.

It should be noted that researchers are divided on what the differences between a norm and a convention are. Gibbs [38, pg. 592] notes that *the terms "convention" and "custom" are frequently employed in the discussions of norms, but there does not* *appear to be any consensus in definitions of them beyond the point that they may not be sanctioned*. We will assume that a convention is a common expectation amongst (most) others that an agent should adopt a particular action or behaviour (e.g. the convention in ancient Rome was to drive on the left). In this paper our view is that norm violations are sanctioned.

Researchers have divided norms into different categories. Tuomela [73] has grouped norms into two categories: social norms and personal norms. Social norms define the behaviour of the group and are associated with sanctions. Personal norms are based on the personal beliefs of the individuals. Personal norms are the potential social norms. These norms may become social norms if they were to be observed by other agents and if sanctions were associated with not following the norm. Social norms are further classified into r-norms (rule norms) and s-norms (social norms). Personal norms are categorised into m-norms (moral norms) and p-norms (prudential norms). Rule norms are imposed by an authority based on an agreement between the members (e.g. one has to pay taxes). Social norms apply to large groups such as a whole society and they are based on mutual belief (e.g. one should not litter). Members of a society expect that a social norm be followed by other members of the society. Moral norms appeal to one's conscience (e.g. one should not steal or accept bribes). Prudential norms are based on rationality (e.g. one ought to maximize one's expected utility). When members of a society violate societal norms, they may be punished or even ostracised in some cases [59].

Many social scientists have studied why norms are followed. Some of the reasons for norm adherence include:

- fear of authority or power [7]
- rational appeal of the norms [2, 9]
- emotions such as shame, guilt and embarrassment that arise because of non-adherence [33]
- willingness to follow the crowd [34]

In this work, we focus on social norms because the agents in multi-agent systems have been modelled using ideas borrowed from social concepts such as speech act theory [66], collaboration and cooperation [56]. Based on the definitions provided by various researchers, we note that the notion of a social norm is generally made up of the following three aspects.

- **Normative expectation of a behavioural regularity**: There is a general agreement within the society that a behaviour is expected on the part of an agent (or actor) by others in a society, in a given circumstance.
- **Norm enforcement mechanism**: When an agent does not follow the norm, it could be subjected to a sanction. The sanction could include monetary or physical punishment in the real world which can trigger emotions (embarrassment, guilt, etc.) or direct loss of utility. Other kind of sanctions could include agents not being willing to interact with an agent that violated the norm or the decrease of its reputation score. Agents that follow the norm might be rewarded.
- **Norm spreading mechanism**: Examples of norm spreading factors include the advice from powerful leaders and entrepreneurs, and the cultural and evolutionary influences. For an external observer, agents identifying and adopting norms

through learning mechanisms such as imitation may also appear to spread norms in agent societies.

2.2 Normative multi-agent systems

Researchers in multi-agent systems have studied how the concept of norms can be applied to software agents. Norms are of interest to multi-agent system (MAS) researchers as they help in sustaining social order and increase the predictability of behaviour in the society. Researchers have shown that norms improve cooperation and collaboration [68, 80]. Epstein has shown that norms reduce the amount of computation required to make a decision [34]. However, software agents may tend to deviate from norms due to their autonomy. So, the study of norms has become important to MAS researchers as they can build robust multi-agent systems using the concept of norms and also experiment on how norms may evolve and adapt in response to environmental changes.

The definition of normative multi-agent systems as described by the researchers involved in the NorMAS 2007 workshop is as follows [15]. *A normative multi-agent system is a multi-agent system organized by means of mechanisms to represent, communicate, distribute, detect, create, modify and enforce norms, and mechanisms to deliberate about norms and detect norm violation and fulfillment.*

Researchers in normative multi-agent systems have been influenced from two different perspectives, philosophy of law (prescriptive approach) and conventionalistic approach (emergence approach) [25]. Based on these two perspectives, research in normative multi-agent systems can be categorized into two branches. The first branch focuses on normative system architectures, norm representations, norm adherence and the associated punitive or incentive measures. The second branch is concerned with the emergence of norms.

Researchers from the philosophical background interested in the study of relationships between different mental attitudes of agents have formalized their theories such as the Belief-Desire-Intention theory[4]. In the study of the prescriptive approach towards norms, researchers are interested in representing norms independent of the domain that is being studied. So, they have used some form of logic, mainly deontic logic to represent norms. Deontic logic is the logic of prohibitions, obligations and permissions [83]. Prohibition norms apply to actions that an agent may perform or the undesired state of affairs that an action may bring about. For example, an agent may be prohibited to litter a park. Obligations are actions that an agent is expected to perform or a state it is expected to bring about. For example, an agent may be obliged to tip a waiter in a restaurant. Permissions are used to indicate exceptions to a general rule or used in cases of uncertainty [31]. For example, a student attending a lecture may be allowed to run out of a lecture theatre in the event of a fire. Deontic logic studies the relationship between these three concepts and also how violations

[4] Beliefs are statements of properties of the world the agent is situated in that can either be true or false; desires are states/situations that an agent prefers to bring about or actions that it wants to perform and intentions are those feasible actions, plans or desired situations that an agent has selected and committed to performing or achieving (c.f. Dignum et al. [32]).

and contrary-to-duty obligations are related [14]. Norms can be represented as rules or conditions using deontic logic. Even though this branch studies how norms are formalized and represented, it does not address the question of where the norm comes from (i.e. how does a norm emerge in a society). Some researchers have proposed mechanisms by which norms can emerge in an agent society [67, 77].

The second branch of research deals with the empirical approaches to norms. The Dictionary.com defines "empirical" as *derived from or guided by experience or experiment* or *provable or verifiable by experience or experiment* [30, adjective, senses 1,3]. This branch of work differs from the first branch in terms of different mechanisms explored by the researchers (e.g. leadership, reputation, machine learning, imitation) and the experiments that are conducted using these mechanisms (refer to [65] for an overview of the mechanisms). Emergence of norms is explored only by research works of this branch. We note that much of the work on norms in this branch does not make any distinction between conventions and norms [7, 63, 67, 68, 78]. Researchers have included both conventions and norms under the umbrella of norms. Most works on the emergence of norms (mainly conventions) are from a game-theory perspective [7, 67, 68].

Conte and Castelfranchi [25] have worked on an integrated view of norms. Their work tries to bridge the gap between the prescriptive view of norms (first branch) and the emergence of conventions (second branch) using the cognitive abilities of an agent. They have proposed a logic-based framework to integrate these two perspectives. However, concrete implementations of this integrated approach are yet to be seen.

2.3 Related work

The work reported in this paper falls under the bottom-up approach in the study of norms. Many researchers in this approach have experimented with game-theoretical models for norm emergence [60, 67, 68]. A necessary condition for a norm to emerge in a society is that the individual agents should first be able identify or recognize the norms in the society. Agents employed by many game-theoretical models learn to choose a strategy that maximizes utility [60, 67, 68]. This utility maximizing strategy is identified by the agents as the norm. However, the agents in these works do not possess the notion of "normative expectation". Hence, norm violations, detection of these violations and norm enforcement are not considered by these works.

Several researchers have proposed architectures for normative systems [13, 18, 32, 49, 61, 76]. Researchers have created these different architectures to study and test their intuitions about norms [54]. These architectures assume that norms exist in the society and the focus is on how the norms can be regulated in an institutional setting. Very few have investigated how an agent comes to know the norms of the society. Our objective in this work is to propose an architecture where agents can identify what the norms of the society are. For a comparison of these architectures refer to Neumann's article [54].

We note that the work that has been carried out by the researchers involved in the EMIL project [4, 5, 50] is related to our work on norm identification. Researchers involved in the EMIL project [5] have worked on a cognitive architecture for norm emergence. They aim to deliver a simulation-based theory of norm innovation, where norm innovation is defined as the two-way dynamics of an inter-agent process and an intra-agent process. The inter-agent process results in the emergence of norms where the micro interactions produce macro behaviour (norms). The intra-agent process refers to what goes inside an agent's mind so that it can recognize what the norms of the society are. This approach uses cognitive agents that examine interactions between agents and are able to recognize what the norms could be. The agents in this model need not necessarily be utility maximizing like the ones in the learning models. The agents in the model will have the ability to filter external requests that affect normative decisions and will also be able to communicate about norms with other agents. Agents just employing learning algorithms lack these capabilities.

The work reported here differs from the work of Andrighetto et al. [4, 5] in three ways. Firstly, in our architecture we have chosen "reaction" or "signalling" to be a top-level construct for identifying potential norms when the norm of a society is being shaped. We note that a sanction not only may imply a monetary punishment, it could also be an action that could invoke emotions (such as an agent yelling at another might invoke shame or embarrassment on another agent), which can help in norm spreading. Agents can recognize such actions based on their previous experience. Secondly, based on association rule mining [21], we propose an algorithm for norm inference, called the prohibition norm identification (PNI) algorithm, which can be adapted by an autonomous agent for flexible prohibition norm identification. To the best of our knowledge, we are the first to employ association rule mining for the identification of norms[5]. Thirdly, we identify two different sets of norms in an agent's mind: candidate norms and identified norms.

Our architecture can be used to identify co-existing norms while many well-known works identify only one norm that exists in the society [7, 34, 43]. Our work also addresses how an agent might be able to dynamically add, remove and modify norms.

3 Motivational scenarios

In order to motivate the technical contribution of this paper, we present the following scenarios from open agent societies.

[5] Some works in the legal domain have investigated the extraction of rules from large databases using association rule mining approach [10,39]. In the work of Bench-Capon et al. [10], rules are identified from large sets of legal description data stored in a database. In a similar vein, Governatori and Stranieri [39] employ association rule mining to extract rules and have used defeasible logic to formally encode the knowledge extracted. Our work is different from these two works employing association rule mining. First, those works identify rules rather than norms. Second, sanctions are not considered in these works. In our work the starting point of norm identification is the recognition of sanctions. Third, the data considered in those works comes from stored data repository which is static, while the data in our work is highly dynamic since it is based on observation of an agent which changes from time to time. Fourth, the notion of distributed agents each inferring norms in our work is unique. These works on the other hand explore rule extraction using a centralized component.

3.1 Scenario 1 - Virtual environments such as Second Life

Divya Chandran, a new Second Life resident wants to explore the rich interactions offered by the new medium. She wants to go to a virtual park and relax by the fountain and listen to chirping birds. She flies to the virtual park and sees people enjoying the sun. She notices some water fountains and some soft-drink fountains from the sponsor of the park. She would like to get a drink, but does not know if there is a norm governing the usage of the fountain. She wonders if she should get a cup from the jazzy sponsor's booth by paying money or if she needs to acquire the skill of making the cup object. Once she fills the cup with the drink, can she enjoy her drink in all areas of the park or is she restricted to a particular zone? And finally, what should she be doing with the empty cup? What is the norm associated with littering in the park? Can she drop it anywhere for an autonomous robot to collect it once the cup is dropped or should she find a rubbish bin and drop it? Will the norm of this park be applicable to all the parks in Second Life? When she visits the park at a later date will the norm of the park still be the same?

3.2 Scenario 2 - E-commerce environments

Rita Ranjan, a casual user of an auction website wants to participate in an auction. She knows the type of auction she is participating in (e.g. English auction, Dutch auction [52]). But she is not aware of the norms associated with the auction. For example, it may not be apparent to her that it is forbidden in a particular auction house to bid for a fourth item of a particular type after winning three consecutive items of the same type (in order to provide a fair chance for others).

The important question the scenarios described above pose is how will the participants (agents) come to know the norms of the society that they currently belong to (or the norms of the context they are in). Knowing the norms is important to the agents, because the agents can protect themselves from negative consequences (e.g. sanctions or a decrease in reputation). In a nutshell, the research question is *what capabilities should be built into an agent to recognize a norm and in particular, prohibition norms?*. The internal agent architecture described in this paper aims to address this question and the experiments described demonstrate how the architecture facilitates the identification of prohibition norms. In particular, it will demonstrate the adaptive behaviour of an artificial agent (e.g. an avatar described in scenario 1 or a software entity (a bot) described in scenario 2) in identifying norms in a society (i.e. adding new norms, recognizing norm changes, and deleting a norm when it does not hold). Note that the society in which the artificial agent is situated can be made up of other autonomous software entities (e.g. bots and avatars).

4 Overview of the process of norm identification

In this section we provide an overview of the norm identification framework for an agent to infer norms in the agent society in which it is situated. Social learning theory

[8] suggests that new behaviour can be learnt through the observation of punishments and rewards[6]. It has been noted that the observation of agent actions through the process of social monitoring and learning [26, 27] can be used to identify norms in the society. An agent employing our architecture makes use of social monitoring and learning approaches to identify norms.

Figure 1 shows the architectural diagram of the norm identification framework of an agent called the norm engine. An agent's norm engine is made up of several components. The circles represent information storage components. The rounded boxes represent information processing components, and diamonds represent decision making components, and the lines represent the flow of information between the components.

An agent employing this architecture follows a four-step process.

Step 1: An agent perceives the events in the environment in which it is situated.

Step 2: When an agent perceives an event, it stores the event in its belief base. Belief base is a repository that contains the beliefs of an agent[7]. An agent has full control of its belief base and thus can add, modify and delete beliefs based on the evidences for these beliefs. The events observed by an observer are of two types: regular events and signalling events. In the context of enjoying a public park, a regular event is an event, such as an agent moving to another location in a park or sitting on a bench. "Special events" are signalling events[8] that agents understand to be either encouraging or discouraging certain behaviour. For example when an agent litters in the park, another agent can discourage the littering action by shouting at the litterer. The signal in this context is the shouting action. We assume that an agent has the ability to recognize signalling events based on its previous experience. Another example of a norm in the context of a restaurant is expectation that the customers should tip the waiter before departing the restaurant. A customer may be sanctioned by the waiter agent. The sanction here could be a *yell* or *shout* action[9]. Sanctions are important because they distinguish a norm from a convention (a behavioural regularity).

Step 3: When a special event occurs, the agent stores the special event in the special events base. It should be noted that all events are stored in an agent's belief base but only special events are stored in the special events base.

[6] In this work we consider only punishments, although the process developed in this work can also be applied to rewards.

[7] The term base is used to represent the repository used by an agent (e.g. an agent's run-time memory or a persistent storage used by an agent such as a database).

[8] Our usage of signalling in this work differs from the views in Biology and Economics. Biologists have observed that animals send signals to indicate that they are a desirable mate [71]. For example, a peacock displays its quality to peahens through its bright plumage and a long ornate tail. Economists have noted that human agents send signals to others that they are credible through some form of signalling (e.g. acquiring a university degree signals that someone has skills for a particular job [72]). In our work, signalling is special type of event whose occurrence can be interpreted as a punishment signal (a sanction). These signals are responses or feedback of the observing agents on the actions performed by the observed agent. While the actions performed by agents themselves are viewed as signals in disciplines such as Biology and Economics, the signals in our work are the feedback of other agents on the actions performed by an agent.

[9] The sanction can also be any other disapproval gesture.

Step 4: If the perceived event is a special event, an agent checks if there exists a norm in its *personal norm* (*p-norm*) base or the *group norm* (*g-norm*) base[10]. An agent may possess some p-norms based on its past experience or preference. A *p-norm* is the personal value of an agent. For example an agent may expect that agents should not litter a park. This personal value may not be shared by the agents in a society. A *p-norm* may vary across agents, since a society may be made up agents with different backgrounds and experiences. A punisher agent may discourage another agent that violates its *p-norm* by punishing it. For example, agent A's *p-norm* might be that no one should litter in the park. If an agent B violates this *p-norm* of agent A, then A may punish B. Agent A can be seen as the norm entrepreneur (i.e. an agent who proposes a norm) and we assume that agent A has the power to punish agent B based on the social roles they play in a society. A *g-norm* is a norm that an agent infers, based on its personal interactions as well as the interactions it observes in the society. An agent infers *g-norms* using the norm inference component of the framework. The focus of this work is on the *g-norm* inference.

When a special event occurs, an agent may occasionally decide to invoke its norm inference component to identify whether a previously unknown norm may have resulted in the occurrence of the special event. In the context of a park scenario, after observing sanctions between agents over certain amount of time (e.g. agents yelling at one another) an agent invokes the norm inference component to find out what past events may have triggered the occurrence of the special event (i.e. sanctioning event). In other words an agent is interested to find out whether the occurrence of a special event can be explained by the existence of a norm in the society. The invocation of the norm inference component may result in the identification of a *g-norm*, in which case it is added to the *g-norm* base. The two sub-components of the norm inference component are the candidate norm identification component and the norm verification component which are introduced in Section 5.2 and further elaborated in Sections 7 and 8 respectively.

The norm inference mechanism can identify two types of norms, prohibition norms and obligation norms. Prohibition norms may be identified by inferring the relevant events that happened in the past (e.g. identifying littering as the action responsible for the occurrence of a sanction). For identifying obligation norms the agent should reason about what events that did not happen in the past are the likely reason for a sanction (e.g. not fulfilling the obligation of tipping in restaurants). The identification of prohibition norms is the focus of this paper. The process of identifying obligation norms has been discussed elsewhere[11].

An agent, being an autonomous entity, can also decide not to invoke its norm inference component for every occurrence of a special event but may decide to invoke it only periodically (e.g. in regular intervals of time[12]). When it invokes the norm inference component, it may find a new *g-norm* which it adds to its *g-norm* base. If it does not find a *g-norm*, the agent may change some of its norm inference parameters

[10] If the event is not a special event, nothing happens. We have only shown the branching conditions which have some consequences in Figure 1.

[11] This paper has not been cited to preserve anonymity.

[12] An agent may collect more evidence about sanctions if it waits for certain period of time.

and repeat the process again in order to find a *g-norm* or may wait to collect more information.

At regular intervals of time, an agent re-evaluates the g-norms it currently has, to check whether those norms hold by invoking the norm inference component[13]. For example, an agent re-evaluates whether a norm it has identified earlier still holds after certain amount of time. If the *g-norm* holds it retains the norm. Otherwise, it removes the *g-norm* from the *g-norm* base. The operational details of the norm inference component are explained in section 5.2. What an agent does with the norms once it has inferred the norms is out of the scope of this paper. Several other works have contributed to strategies that agents can follow to decide whether to follow a norm [29,48,53]. An assumption these works make is that the agents are aware of the norm ahead of time. The present work aims to provide a solution towards bridging the gap of how an agent infers the norm.

When an agent invokes its norm inference component and finds that a *g-norm* does not apply (e.g. if it does not find any evidence of sanctions), it deletes the norm from the *g-norm* base. There could be two reasons why there are no sanctions in the society. The first reason is that there may not be any punishers in the society. When there are no punishers (i.e. norm guardians), agents can delete the norm. Second reason is that all the agents may have internalized the norm (accepted the norm) and are following the norm[14]. Hence, there might not be any sanctions. In this case, the norm deletion on the part an agent who has not internalized the norm may have negative consequences for that agent (i.e. the agent can be sanctioned) in which case it can add the norm again through norm inference. Agents may delete norms that apply to a society when they leave a society[15].

5 Overview of the components of the norm identification framework

This section provides an overview of the components of the norm identification framework. The components that will be discussed are a) event storage components and b) the norm inference component. We will describe the role of the components in the context of a park scenario.

[13] In human societies norm re-evaluation happens rarely as the norms tend to be largely permanent (e.g. the norms of cooperation and reciprocity). However, some social norms may change (e.g. smoking in a restaurants which was originally permitted is now prohibited) that require re-evaluation on the part of agents. We believe, in virtual environments, the norms can change at a faster rate as the composition of an open agent society changes. Hence, agents need to re-evaluate norms in regular intervals of time. We note that an agent can modify how often it re-evaluates norms using a parameter which can be used to model either frequent or rare re-evaluations.

[14] Note that how an agent internalizes a norm is out of the scope of this work. Other researchers have studied how norms are internalized [77]. The focus of our work is on norm identification.

[15] We have experimented with two types of scenarios where agents remember the norms of the society they leave and where agents remove the norms of the society they leave.

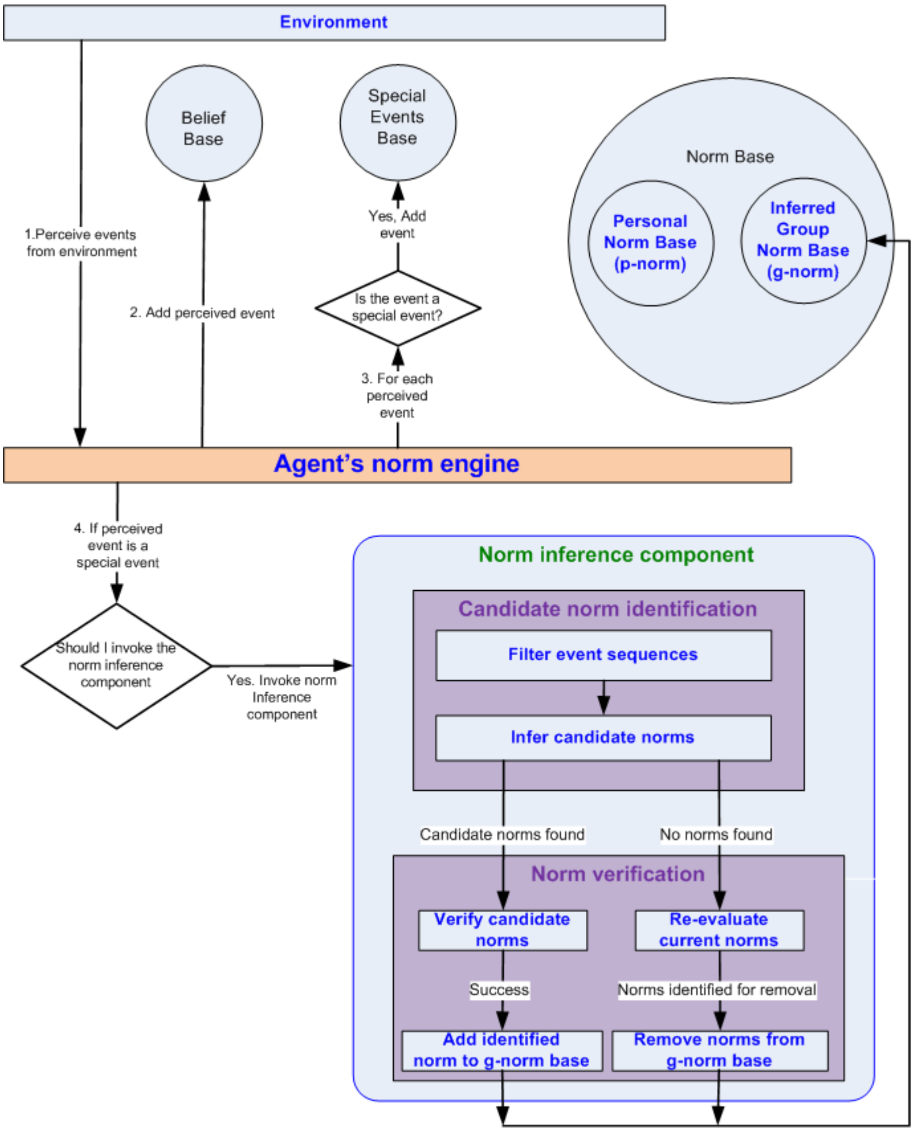


Fig. 1 Architecture of the norm identification framework of an agent

5.1 Event storage components

Let us assume that an agent is situated in a public park. The agents are aware that they are in a park, and interactions happen within the park context. Additionally, each agent knows other related information about the environment, such as the location of the rubbish bins. Let us also assume that a norm against littering does not exist to start with, but a few of the agents (sanctioning agents) have a notion of what an appropriate action should be in a particular circumstance (a *p-norm*). In this architecture an agent would first observe the interactions that occur between the agents in the society. The interactions could be of two types. The first type of interaction is the one in which the agent itself is involved and is called a *personnel interaction* (an action that an agent does in an environment or a message that is exchanged with another agent). The second type of interaction is an interaction between other agents that is observed by an observer agent, referred to as an *observed interaction*. The agent records these interactions (events) in its belief base. An agent in the society can assume one or more of the three simple roles: a participant (P) that is involved in a personal interaction, an observer (O) that observes what other agents do and a signaller (S) who sanctions other agents either because the action is against its *p-norm* or is against the *g-norm* of a group that it previously belonged or currently belongs. We assume that the signalling agent has the social power [12,45] to sanction other agents similar to what is observed in human societies (e.g. a jaywalker being reprimanded, a litterer being rebuked by a member of the public). These signals can either be verbal (e.g. yelling) or non-verbal (e.g. shaking head in disapproval).

When the agents move around the park and enjoy the environment, they may become hungry and eat food. Some agents may litter (i.e. drop the rubbish on the ground), and some agents may dispose rubbish in a rubbish bin. The actions that can be performed by an agent are *move*, *eat*, and *litter*. Some agents consider littering to be an activity that should be discouraged (based on their p-norms or past experience), so they choose to signal other agents through actions such as *yelling* and *shaking* their heads in disapproval. We assume that an agent has a filtering mechanism which categorizes actions such as *yell* and *shake-head* as signalling actions. Signalling events are negative events (sanctions). These signalling actions are stored in the special events base. The signalling agents can be considered as norm proposers.

We assume that agents can observe each other within a certain visibility threshold (e.g. agents can only see other agents in a certain neighbourhood in a grid environment). An observer records another agent's actions until it disappears from its vicinity. When a sanctioning agent observes the occurrence of an event that is in breach of one of its norms, the agent may become emotionally charged and perform a sanctioning action, such as shaking its head vigorously in disapproval. The observer agent observing this interaction can infer that someone involved in an interaction may have violated a norm. Even though an observer may know that a sanctioning event has occurred, it may not know the exact reason for sanctioning (i.e. it may not know the norm). It will infer norms using the norm identification framework.

In order to understand what an agent stores, let us assume that an agent perceives other agents' actions. An event that is perceived consists of an event index, an observed action, and the agent(s) participating in that event. For example an agent observing another agent eating will have the representation of *happens(1,eat(A))*. This implies the observer believes that at time unit one, agent A performs an action *eat*. A sample representation of events observed by an agent is given in listing 1.

$$\begin{pmatrix} happens(1, eat(A)) \\ happens(2, litter(A)) \\ happens(3, move(B)) \\ happens(4, move(A)) \\ happens(5, sanction(B, A)) \end{pmatrix} \tag{1}$$

An agent records these events in its belief base. Event 5 is a sanctioning event, where agent B sanctions agent A. The agents have a filtering mechanism, which identifies signalling events and stores it in the special events base. It should be noted that special events, such as *yell* and *disapproval shake*, are categorized by an agent as sanctioning events and they are stored in the special events base under the *sanction* event. We note that recognizing and categorizing an event into a sanction is a difficult problem. In our architecture we assume such a mechanism exists (e.g. based on an agent's past experience). Several researchers are working on detecting emotions in artificial agents such as avatars through extracting facial features, messages and emoticons exchanged between participants [46, 55, 58]. We assume that a module that infers emotions already exists (e.g. one of the above mentioned approaches may be employed to recognize and categorize events).

5.2 Norm inference component

The norm inference component of an agent is made up of two sub-components. The first sub-component makes use of our Prohibition Norm Identification (PNI) algorithm to generate candidate prohibition norms. The second sub-component is the norm verification component, which verifies whether a candidate norm can be identified as a norm in the society.

5.2.1 Candidate norm identification component

Candidate norms are the norms that an agent considers to be potential candidates to become the norms in a society. The prohibition norm identification mechanism identifies candidate prohibition norms. Identifying candidate norms is a three step process (Figure 2). First, based on the observations an agent creates event sequences. These are sequences of events that happen between interacting agents. Second, the agent filters event sequences based on a criterion. The criterion is that the agent is interested only in events that precede a sanction in order to identify a norm. Third it uses a data mining approach [21, 42] to identify candidate prohibition norms. The identified prohibition norms are stored in candidate prohibition norm set (CPNS).

A higher level overview of identifying prohibition norms is given in Figure 2. The steps involved are represented using rounded rectangles in Column one. Columns two and three show the state information associated with identifying prohibition norms. The state information includes the resultant products that are created (solid rectangles), and the processes (circles). Each of these three steps produce certain products (or results). The product of a step is used by the subsequent step. A detailed description of these products and processes associated with the identification of candidate prohibition norms is provided in Section 7.

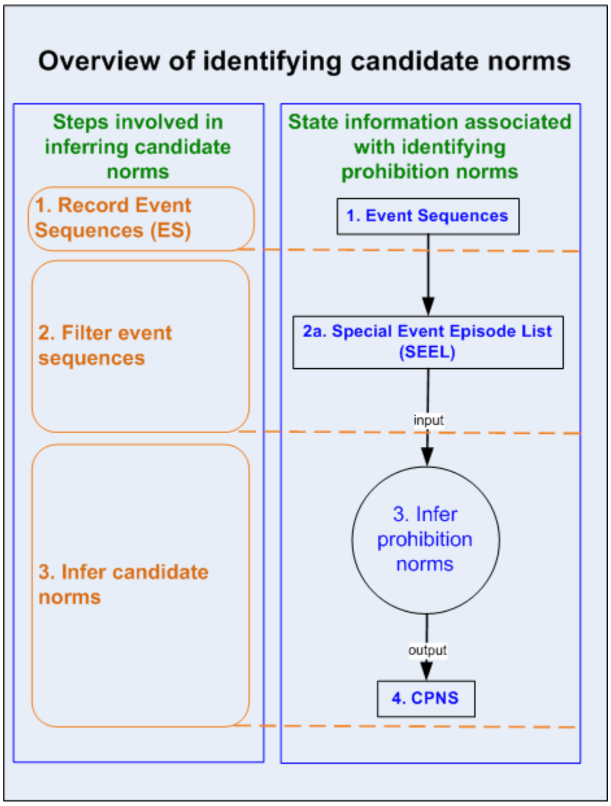


Fig. 2 Overview of the norm inference component

5.2.2 Norm verification component

Once the candidate norms have been obtained, the norm verification component verifies whether a candidate norm can be identified as a norm in the society. An agent asks other agents in its vicinity whether a particular norm holds in the society. If the answer is yes, it then adds the norm to its g-norm base. If the answer is no, then the agent does not add the norm. If the other agent does not know whether the norm holds the agent may ask other agents or choose to gather more data.

In our architecture a new agent entering the society does not ask another agent what the norms of the society are. It first uses the norm identification architecture to infer the norms. It then asks for norm verification[16]

Detailed description of these two components (candidate norm identification and norm verification) are presented in the context of identifying prohibition norms in Sections 7 and 8, respectively.

6 Definitions of attributes of the norm identification framework

The attributes (or the parameters) of the norm identification framework are defined below. The list of all acronyms used in this paper and their corresponding expansions are provided in Table 3 in the appendix.

History Length (HL): An agent keeps a history of the observed interactions for a certain window of time, which is represented by the History length (HL) parameter. For example, if HL is set to 20, an agent will keep the events it observes in the last 20 time steps in its memory.

Event Sequences (ES): An event sequence is the record of actions that an agent observes in the history. For example the event sequence observed by an agent where HL=5 is given in Table 1. In this case, the observer agent observes two other agents, A and B.

Special Events Set (SES) : An agent has a set of events it identifies to be special. These events are the signalling events. For example, the special event set can contain events such as *yell*, or *shake head in disapproval* (SES = { *yell*, *disapproval head-shake* }). We assume that an agent has the capability to categorize events into sanctions. For example the actions mentioned above can be identified as sanctioning actions.

Unique Events Set (UES): This set contains the number of distinct events that occur within a period of time. For example, a unique events set for the in a park may

[16] The reasons for asking another agent just for verification are two-fold. First, an agent entering a society may not be interested to find out all the norms of a society (an agent might give a long list of norms followed in the society). It may be interested to find the norms in a particular context. An agent has to first infer what the context is (by observing the interactions) and then it can ask another agent in the neighborhood if its inference of a norm is valid (e.g. *Am I obliged to tip in this society?*). In our view this is more effective (in terms of computation and memory required) than asking another agent what the norms are, as there could be a long list of norms that apply, and most of those may not be of interest to an agent. The agent employing the architecture will be able to infer what the potential norms might be. Hence, it can be confident in asking for norm referral, as the actual norm might be one of the candidate norms in its list. The search space for the actual norm has been narrowed by the norm identification process. An agent can also precisely formulate a query for another agent (e.g. *Is it prohibited to litter in this society?*).
Second, an agent may not completely trust other agents in an open society. When an agent asks another agent without norm inference, the other agent could potentially lie about a norm [64]. So, an agent may want to make sure that it identifies candidate norms, before it asks for norm verification. This process helps an agent from being misled by the referring agent if it were to ask what the norm is, since it knows that one of the candidate norms could potentially be a norm. Note that this does not solve the lying problem, since the referrer agent can lie when an agent asks if something is a norm in the society. At the least, the mechanism we use here allows the agent to have a set of candidate norms. We discuss a potential solution to the lying problem in Section 10.

contain the following events, UES = $\{$ *eat, litter, move, sanction* $\}$. Note that the event occurrences are modelled as simple propositions.

Occurrence Probability (OP): The occurrence probability of an event E is given by the following formula.

$$OP(E) = \frac{\text{Number of occurrences of E}}{\text{Total number of events in ES}}$$

Window size (WS): When an agent wants to infer norms, it looks into its history, a certain number of recent events that precede a sanction. For example, if the WS is set to 3, an agent constructs an Special Event Episode (SEE) with three events that precede a special event. Construction of event episodes is described in Section 7). It should be noted that an SEE is a subset of ES.

Norm Identification Threshold (NIT): When coming up with candidate norms, an agent may not be interested in events that have a lower probability of being a norm. For example, if an agent sets NIT to be 50 (in a scale from 0 to 100), it indicates it is interested to find all sub-episodes of an event episode that have 50% chance of being a candidate norm (i.e. being the reason for generating a sanction).

Norm Inference Frequency (NIF): An agent may choose to invoke a norm inference component every time it observes a special event, or may invoke the component periodically. An agent has a parameter called norm inference frequency (NIF) that specifies what the time interval between two invocations of the norm inference component is. An agent, being an autonomous entity, can change this parameter dynamically. If it sees that the norm in a society is not changing, then it can increase the waiting period for the invocation of the norm inference component. Alternatively, it can reduce the time interval if it sees the norm is changing.

7 Candidate norm identification

There are two main steps involved in the process of inferring candidate norms is given in Algorithm 1. First, based on a filtering mechanism, special event episodes are extracted from the event sequences that are recorded by the agent. Second, based on the special event episodes that are extracted, the Candidate Prohibition Norm Set (CPNS) is generated using a modified version of the WINEPI algorithm [51].

Algorithm 1: Overview of the norm inference process

foreach *invocation of the norm inference component* **do**
    Create special event episodes from event sequences; `/* see Algorithm 2 */`
    Create candidate prohibition norms; `/* see Algorithm 3 */`
end

7.1 Creating special event episodes

An agent records other agents' actions in its belief base. We call these events that were recorded in the belief base "event sequences" (ES). An agent has a certain history length (HL). Let us assume that there are three agents A, B and C. Agent C observes agents A and B interacting with each other. A sample representation of events observed by agent C is given in Table 1. We assume that agents can perform only one action at any point in time. It can be observed that agent A eats in time unit 1, litters in time unit 2, and then moves for the next three time units, while agent B eats in time unit 1, moves for the next two time units, sanctions agent A in time unit 4 and then moves.

Table 1 Events observed by an agent

Agent A	Agent B
happens (1,eat(A))	happens (1,eat(B))
happens (2,litter(A))	happens (2,move(B))
happens (3,move(A))	happens (3,move(B))
happens (4,move(A))	happens (4,sanction(B,A))
happens (5,move(A))	happens (5,move(B))

When an agent observes a special event (e.g. sanction), it extracts the sequence of actions from the recorded history (event sequences) that were exchanged between the sanctioning agent and the sanctioned agent. In the example shown in Table 1, the observer infers that something that agent A did may have caused the sanction. It could also be something that agent A failed to do might have caused a sanction. In this paper we concentrate on the former possibility (i.e. the identification of prohibited actions). Agent C then extracts the following sequence of events that took place between A and B.

$$\{A,B\} \rightarrow happens(1,eat(A)) - happens(2,litter(A)) - \\ happens(3,move(A)) - happens(4,sanction(B,A))$$

We call the retrieved event sequence that precedes a sanction as the *special event episode* (SEE). To simplify the notation, only the first letter of each event will be mentioned from here on (e.g. *e* for eat). Thus the event episode for interactions between agents A and B shown above will be represented as

$$\big(\{A,B\} \rightarrow e - l - m - s\big)$$

There might be a few sanctioning events at any given point of time that an agent observes. A sample special event episode list (SEEL) that contains events that are observed by an agent preceding a sanction where WS=3 is given in Figure 3.

$$\begin{pmatrix} e-l-m-s, l-e-l-s, m-e-l-s, e-l-e-s, e-l-e-s \\ l-e-l-s, e-e-l-s, m-e-l-s, e-l-m-s, e-l-e-s \end{pmatrix}$$

Fig. 3 A sample special event episode list (SEEL)

The pseudocode for creating an special event episode list (SEEL) is given in Algorithm 2. For every special event in the event sequence (ES), an agent creates special event episode (SEE) with *n* events that precede the special event where, n=WS. Special event episode (SEE) is then added to the special event episode list (SEEL). Even though in the example shown in Figure 3, we have assumed that an agent considers three events (*n*=3) that precede a signal (a sanction), the value of *n* can change according to the computational capabilities of the agent.

Algorithm 2: Pseudocode to create Event Episode List

```
Input: Event Sequence (ES), Window Size (WS)
Output: Special Event Episode List (SEEL)
foreach invocation of the norm inference component do
    foreach special event in ES do
        Create special event episode (SEE) from the last n events that precede
        the special event, where n=WS;
        Store special event episode (SEE) in special event episode list (SEEL);
    end
end
```

7.2 Generating candidate prohibition norms

The pseudocode for generating the candidate prohibition norms is given in Algorithm 3. Algorithm 3 is a modified version of the WINEPI algorithm [51], an association rule mining algorithm. Association rule mining [21] is a well known field of data mining where relationships between items in a database are discovered. For example, rules such as X% of people who bought diapers also bought beers can be identified from a database. There are several well known algorithms that can be used to mine interesting rules such as Apriori [1] and WINEPI [51]. The WINEPI algorithm analyses event sequences and identifies frequently occurring episodes in a particular window of time.

As norms in this work are inferred from sequences of events that precede a signalling event, we have used a modified version of the WINEPI algorithm to identify candidate norms. For example an observer may observe that a littering action always happens (occurrence probability = 1) before the occurrence of a sanctioning event. In this case, the WINEPI algorithm can be used to identify the relationship between the littering action and the sanctioning action. We have modified the WINEPI algorithm such that it generates sub-episodes using “permutation with repetition”. The pseudo code of the modification is given in algorithm 4.

Algorithm 3: Pseudocode to create Candidate Prohibition Norms Set (CPNS)

Input: Special Event Episode List (SEEL), Unique Event Set (UES), Window Size(WS), Norm Inference Threshold(NIT)

Output: Candidate Prohibition Norms Set (CPNS)

```
1  begin
2      CPNS ⟵ ∅;
3      iterNum ⟵ 1;
4      Sub-Episode List (SEL_current) ⟵ UES;
5      while iterNum ≤ WS do
6          SEL_temp ⟵ ∅;
7          foreach Sub-Episode(SE) in SEL that appears in SEEL do
8              if iterNum = 1 then
9                  if OP(SE) ≥ NIT then
10                     CPNS ⟵ SE, SEL_temp ⟵ SE;
11                 end
12             end
13             else
14                 if each event in SE ∈ CPNS and OP(SE) ≥ NIT then
15                     CPNS ⟵ SE, SEL_temp ⟵ SE;
16                 end
17             end
18         end
19         Construct SEL_next using SEL_temp ;        /* Algorithm 4 */
20         if iterNum < WS then
21             iterNum ⟵ iterNum + 1;
22             SEL_current ⟵ SEL_next;
23         else
24             return CPNS;
25         end
26     end
27 end
```

Algorithm 3 works iteratively, with the number of iterations equal to Window Size (WS). The sub-episodes for the first iteration are of length one. The sub episode list (SEL) for iteration one contains all the events in the UES. For example, for the events in Figure 3, the SEL at the first iteration will contain events *e*, *l* and *m*. For each of the sub-episodes in the SEL, the occurrence probabilities are calculated. If the occurrence probability of a sub-episode in the special event episode list is greater than or equal to the norm inference threshold (NIT), the event is added to the Candidate Prohibition Norms Set (CPNS) (lines 7 to 12). For example, if the occurrence probabilities of events *e* and *l* are greater than or equal to NIT, then these will be added to CPNS. Each candidate norm is also added to a temporary list which is used for creating the SEL for the next iteration. The SEL for the next iteration (SEL_{next}) is created using Algorithm 4 and is assigned to $SEL_{current}$.

Each sub-episode in the second iteration will have two events. In the second iteration a sub-episode in SEL will be added to CPNS if two conditions are satisfied (lines 13 to 17). To be added to CPNS,

1. Each event in the sub-episode should already exist in the CPNS.
2. The occurrence probability of the sub-episode in the special event episode list (SEEL) should be greater than or equal to NIT.

In a similar fashion, the algorithm computes all candidate norms. The maximum length of a candidate norm and the number of iterations of algorithm 3 are also equal to WS.

Algorithm 4: Pseudocode to create Sub-Episode List

Input: Candidate norms list (tempList)
Output: Sub Episode List (SEL)
foreach *candidate norm in tempList* **do**
    Generate Sub-Episodes (SE) of length n using other candidate norms (allowing repetition of events);
    Add each generated SE to SEL;
end

Let us consider the following scenario that demonstrates how the norm identification mechanism works. Assume that an agent is interested in *three* events in an event sequence that precede a sanction (i.e. event episodes of length three). Let us assume that NIT is set to 50%, and the unique event set is $\{e,l,m\}$. As an example let us consider the special event episode list (SEEL) given in Figure 3. In the first iteration, $SEL_{current}$ contains sub-episodes of length one which are *e, l* and *m*. For each sub-episode an agent calculates the occurrence probability. The occurrence probabilities for *e* and *l* are 100% and the occurrence probability for *m* is 40%. Since events *e* and *l* have an occurrence probability greater than NIT, the agent adds these two events to its candidate norm set. For the second iteration, the agent must calculate sub-episodes of length two that have NIT greater than 50%. It uses Algorithm 4 to calculate the next sub-episode list (SEL_{next}). For this purpose it uses the candidate norms that were found in the previous iteration (SEL_{temp}). SEL_{temp} in this case contains $\{e, l\}$.

Algorithm 4 creates sub-episodes for the subsequent iteration based on the candidate norms from the previous iteration. In the running example, in the second iteration the algorithm creates sub-episodes of length two, based on sub-episodes of length one. Allowing repetition of events, the algorithm creates the following sub-episodes $\{ee, el, le, ll\}$ and adds them to the SEL_{next}[17]. Then, the probabilities of these 4 sub-episodes are calculated. Occurrence probabilities of $\{$ *ee, el, le, ll* $\}$ are $\{$ 10%, 100%, 50%, 0% $\}$. As NIT is set to 50%, *el* and *le* are added to the candidate norms set. These two sub-episodes will be considered for the creation of SEL for the third iteration using Algorithm 4. For the third iteration the contents of the SEL are $\{ele, lel\}$.

[17] Permutations with repetitions are considered, because an agent does not know whether littering once (*l*) is a reason for sanction or littering twice (*ll*) is the reason for the sanction. It could be that littering once may be allowed but an agent littering twice may be punished.

In iteration three, the occurrence probabilities of {*lel, ele*} are {20%, 30%}. As the occurrence probabilities of the sub-episodes of length three are below NIT these events will not be added to the candidate norms set. As the number of iterations is equal to WS, the algorithm returns the candidate norm set to the agent. In the end, the candidate prohibition norms set will have the following entries whose occurrence probabilities are greater than or equal to NIT: {*e, l, el, le*}. If the agent sets the NIT to 100% then the CPNS will contain *e, l* and *el*. Table 2 shows the norms identified at the end of different iterations when the Norm Identification Threshold (NIT) is varied based on the sample special event episodes given in Figure 3. It can be noted that the number of candidate norms decrease when the NIT increases.

Table 2 Candidate prohibitions norms on varying Norm Identification Threshold

Iteration	NIT=25%	NIT=50%	NIT=100%
1	e, l, m	e, l	e, l
2	le, el	le, el	el
3	ele	-	-

It should be noted that algorithm 3 is a modified version of the WINEPI algorithm [51], an algorithm for mining association rules. Some well known algorithms in the data mining field can be used for mining frequently occurring episodes (i.e. mining association rules) [1,51]. A limitation of the well-known Apriori [1] algorithm is that it considers combinations of events but not permutations (e.g. it does not distinguish between event sequences *el* and *le*). WINEPI [51] addresses this issue, but it lacks support for identifying sequences that are resultants of permutations with repetition (e.g. from sub-episodes of length one, e.g. *e* and *l*, the algorithm can generate sub-episodes of length two which are *el* and *le*, but not *ee* and *ll*).

The modification we have made to the WINEPI algorithm is given in Algorithm 4 which can identify candidate norms that are obtained by considering “permutations with repetition” when constructing sub-episodes. We note that algorithm 3 can be replaced with any other association rule mining algorithm. Hence, it forms the replaceable component of the framework.

Having compiled a set containing candidate norms, the agent passes this information to the norm verification component.

8 Norm verification

In order to find whether a candidate norm is a norm of the society, the agent asks another agent in its proximity. This happens periodically based on NIF (e.g. once in every 10 iterations).

When two agents A and B interact, A chooses its first candidate norm (say *el*) and asks B if it knows whether *el* is a norm of the society. If the response is affirmative, A stores this norm in its set of *identified norms*. If not, A moves on to the next norm in its candidate norm set[18].

In the case of the running example, the sub-episode *e* has the highest probability for selection, and it is chosen to be communicated to the other agent. It asks another agent (e.g. an agent who is the closest) whether it thinks that the given candidate norm is a norm of the society. If it responds positively, the agent infers $prohibit(e)$ to be a norm. If the response is negative, the next candidate norm set is chosen for verification. The agent then asks whether *l* is the reason for sanction. If yes, littering is considered to be prohibited. Otherwise, the agent moves on to the next candidate norm. This process continues until a norm is found or no norm is found, in which case the process is re-iterated once a new signal indicating a sanction is generated. When one of the candidate norms has been identified as a norm of the society, the agent still iterates through the candidate norm set to find any co-existing norms. For example, assume that two norms exist in a park, a norm against littering and a norm against eating. Having identified a norm against littering, an agent will iterate through its candidate norm set to find the norm against eating.

Note that an agent will have two sets of norm repositories in its memory: candidate norms and identified norms. Figure 4 shows these two sets of norms. Once an agent identifies the norms of the system and finds that the norms identified have been stable for a certain period of time, it can forgo using the norm inference component for a certain amount of time (based on the norm inference frequency (NIF)). It invokes the norm inference component periodically to check if the norms of the society have changed, in which case it replaces the norms in the identified set with the new ones or deletes the norms which are no more applicable. For example, if an agent identifies that the norm against eating does not hold in a society (i.e. there are no sanctions in the society for eating), it removes this norm from its identified norm set[19].

9 Experiments on norm identification

In this section we demonstrate how the agents that make use of the proposed architecture are able to infer the norms of the society.

We have implemented a Java-based simulation environment to demonstrate how norms can be identified by agents in a multi-agent society. A toroidal grid represents a social space where agents can move[20]. The snapshot given in Figure 5 shows different

[18] Other alternative mechanisms are also possible. For example, an agent can verify whether its candidate norms hold by undertaking actions that it observes to be sanctioned (e.g. dropping a litter). Based on the outcome of tests the agent carries out it can infer what the norms could be. This is a meta-level norm testing mechanism of an agent.

[19] It could be that there are no sanctions in the society because all the agents have internalized a norm (i.e. they abide by the norm). However, the process of norm internalization has not been considered in this work. Norm internalization can depend upon the personality of agents (e.g. social agents, rebellious agents). We note this can be included in our architecture. Additionally, norm internalization is a separate issue from the issue of norm identification studied here.

[20] A toroidal grid is constructed from a rectangular grid (shown in Figure 5) by gluing both pairs of opposite edges together. This forms a three dimensional space (a donut shape) where agents move in circles [82].

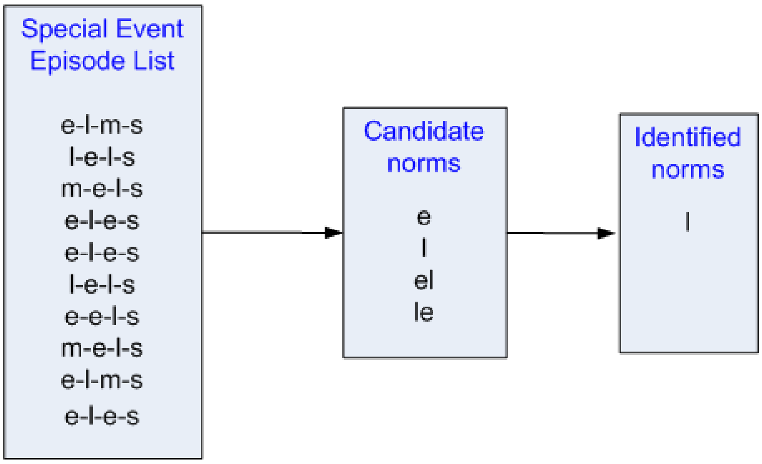


Fig. 4 Two sets of norms

types of agents in a grid environment populated with four different societies. An agent enjoys the park by moving from one location to another. An agent can move in one of the four directions (up, down, left and right). The agents are represented as circles. There are three types of agents in the system: the litterers (L) in light blue, the non-litterers (NL) in light green and the non-littering punishers (NLP) in red. An agent's visibility is limited to a particular society (i.e. an agent can observe the actions of all the other agents in its society). The letters that appear above an agent specify the agent number and the action it is currently performing. When an agent infers a norm, a solid square appears inside the circle with the same colour as that of the signalling agent (red in this case). The signalling agent is a norm proposer which punishes other agents probabilistically based on its *p-norm*. For experiments reported in Sections 9.1 to 9.5, all the agents make use of the norm inference mechanism.

9.1 Experiment 1 - Norm identification and verification

The objective of this experiment is to demonstrate that agents that use the norm inference architecture can generate candidate norms and also identify norms through the verification process. Agents in a society can verify that a certain norm holds in a society by asking other agents in the society. There were 100 agents in the agent society (50 NL, 46 LL and four NLP agents). The NLP agents punished agents that littered.

9.1.1 Norm identification

In order to demonstrate that the norm identification component works, we conducted experiments by varying the NIT and keeping all the other parameters constant (HL=20, NIF=5, WS=3). For a particular agent, when NIT was set to 25%, the agent inferred seven candidate norms $\{e, el, l, le, lel, ee, eel\}$ whose occurrence probabilities were

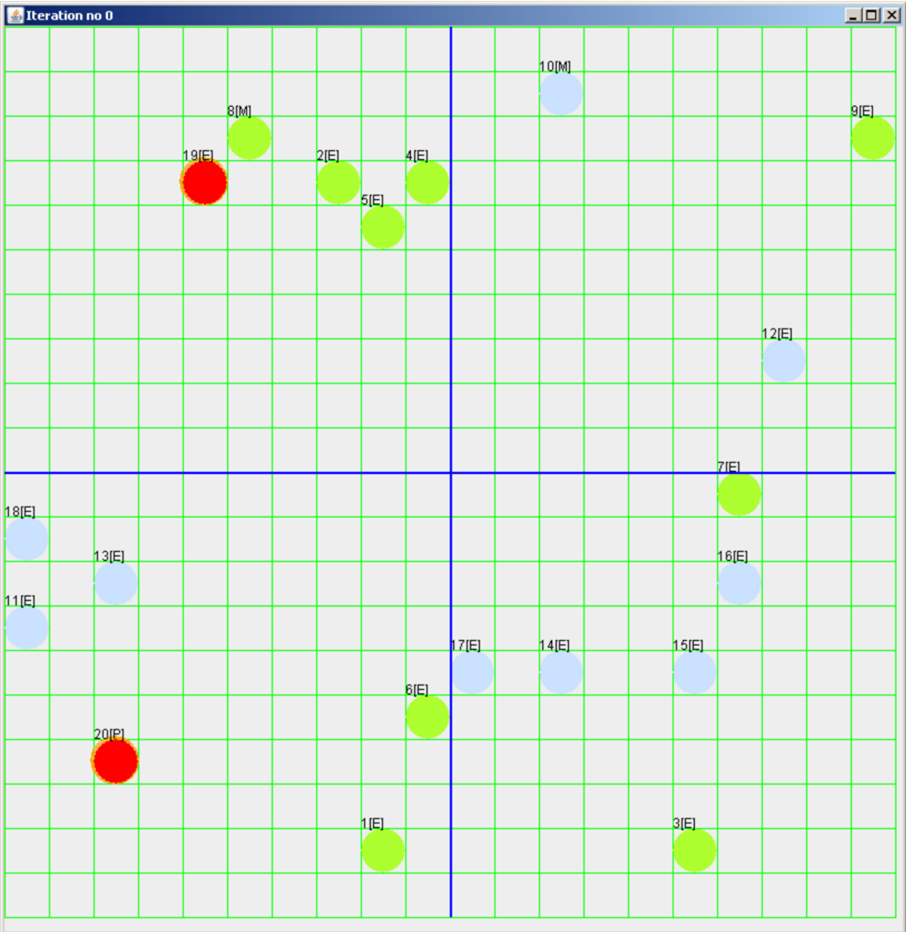


Fig. 5 Snapshot of the simulation environment with 20 agents - the litterers in light blue, the non litterers in light green and the non-littering punishers in red

{*1, 1, 1, 0.75, 0.75, 0.25, 0.25*}. When NIT was set to 50%, the agent inferred five candidate norms {*e, el, l, le, lel*}. Note that the candidate norms that are identified when NIT was set to 50% are a sub-set of the candidate norms that were identified when NIT was set to 25%. When NIT was set to 100%, the agent inferred three candidate norms {*e, el, l*}.

9.1.2 Norm verification

In our experimental set-up an agent can ask one other agent in its vicinity (randomly chosen) about a candidate norm. If that agents answers positively, then the agent will promote the norm to the identified norm set. When seeking norm verifications, an agent can use one of the following approaches. It can either ask a) a sanctioning agent or b) any agent that possesses a norm (e.g. the agent may have obtained the norm from a sanctioning agent)[21]. Figure 6 shows two lines which correspond to these approaches (a sample run where all the other parameters are the same). It can be observed that all the agents in the society identify a norm by iteration 91 if they ask only the punisher, and the same agent society identifies the norm in iteration 7 if it an agent can ask any other agent for norm verification. As the probability of the other agent being a non-punishing agent is higher than being a punishing agent (0.96 vs. 0.04), the norm identification is faster when any agent that has a norm can recommend the norm to other agents (approach b) when compared to norm recommendation only by the sanctioning agents (approach a). In approach b, initially the norms are verified only by the punishers (i.e. the other agents do not know the norm). As more and more non-punishers come to know of the norm from punishers, they can also participate in the norm verification process. This leads to the faster convergence using approach b.

This simple experiment (approach b) reflects what happens in human societies. The norm not only spreads just by the enforcer but also through agents that have been punished or an agent who knows about someone who has been punished[22]. An agent entering a new society asks other agents in a society whether a suspected norm currently holds in a society.

We also conducted experiments with two types of punishers in the society, one that punishes eating activity and the other that punishes littering. The results were similar to the one shown in Figure 6 (i.e. asking any agent converges faster than asking just the punisher). Note that the experiments reported in the rest of the paper use norm recommendations from any agent (approach b).

9.2 Experiment 2 - Dynamic norm change (from society's view point)

Agents in a society should have the ability to infer norms when norms change. A new norm can be established when a punisher leaves the society and a new punisher joins the society or when a punisher agent comes up with a new norm replacing the old one. This experiment demonstrates that agents in different societies can change norms based on inference.

We experimented with four different societies with a total of 50 agents (24 NL, 22 LL and 4 punishers). Out of the four punishers, two agents punished littering actions, and two others punished eating actions in the park. These punishers were randomly assigned to different societies and the punishers are able to move from one society to another[23]. The simulation set up is divided into 4 blue rectangles, and each rectangle represents a distinct society. The agents are represented as coloured circles (see Figure 7). The littering agents are in light blue, while the non-litterers are in light

[21] We note that other mechanisms are also possible. For example, an agent can ask certain number of agents in its vicinity instead of asking just one. In this work, agents ask one other agent for norm verification. There is a parameter in the system which can be used to change the number of agents that are asked for verifying a norm. This is akin to the referral process which has been studied by other researchers [20, 84] which show that increasing the number of referrals leads to faster convergence.

[22] This information can be obtained either through gossip [17, 57] or common knowledge [22].

[23] The simulation of this scenario can be viewed http://unitube.otago.ac.nz/view?m=3d8h11fu1xa

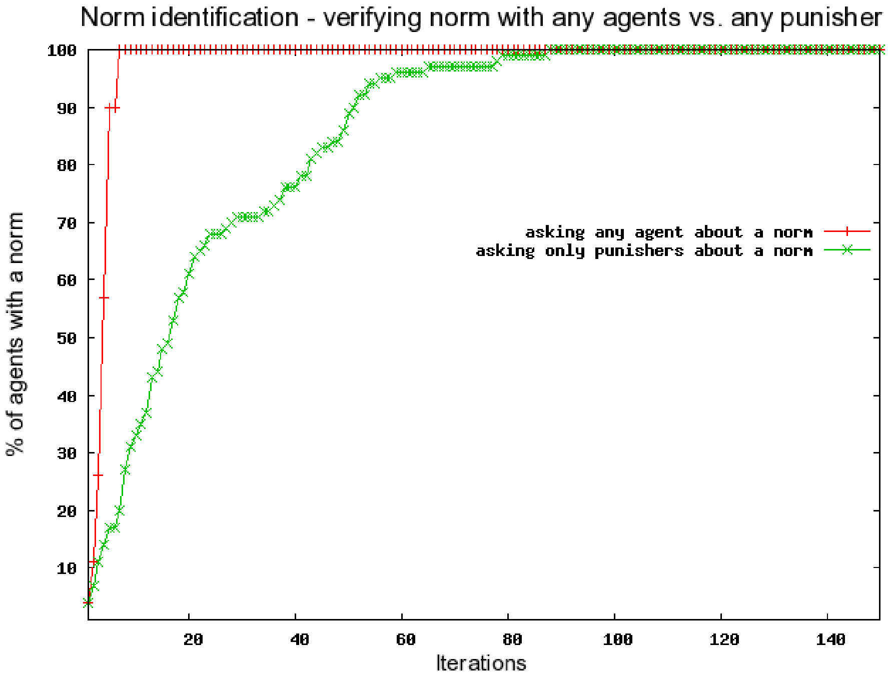


Fig. 6 Comparing the impact of two modes of norm verification (enquiring about a norm to a punisher vs. any agent in the society) on norm convergence in an agent society

green. The punishers that punish littering actions are in red, and the punishers that punish the eating actions are in blue. A small rectangle that appears inside an agent represents the agent infering the norm. The color of the small rectangle corresponds to the punisher's color (hence it corresponds to the identified norm). Figures 8 and 9 show the percentage of agents that have identified a norm in different societies as the punishers dynamically move across societies. The agents that punished litterers punished 25% of the punishing actions, while the agents that punished eating actions punished only 5% of the eating actions.

Figure 8 shows how norms in society 3 (rectangle in the lower-left of the simulation window shown in Figure 8) change dynamically. When the simulation starts, there are both types of punishers in society 3, so agents start infering both types of norms. About iteration 440, the punisher that punishes the eating action moves to society 4 (rectangle in the lower-right of the simulation window), and about iteration 463 the punisher that punishes littering action moves to society 4. As society 3 does not have any punisher, both the norms disappear from this society by iteration 560. The norms disappear because of two reasons. The first reason is that some agents with norms gradually move to other societies and the agents that come to the society may not know the norm to start with. In the absence of the punishers, these new agents do not infer the norms. This gradually reduces the number of agents with a norm in the society. The second reason is that some agents that know the norms may re-evaluate the norm in certain intervals of time (by invoking the norm inference component)

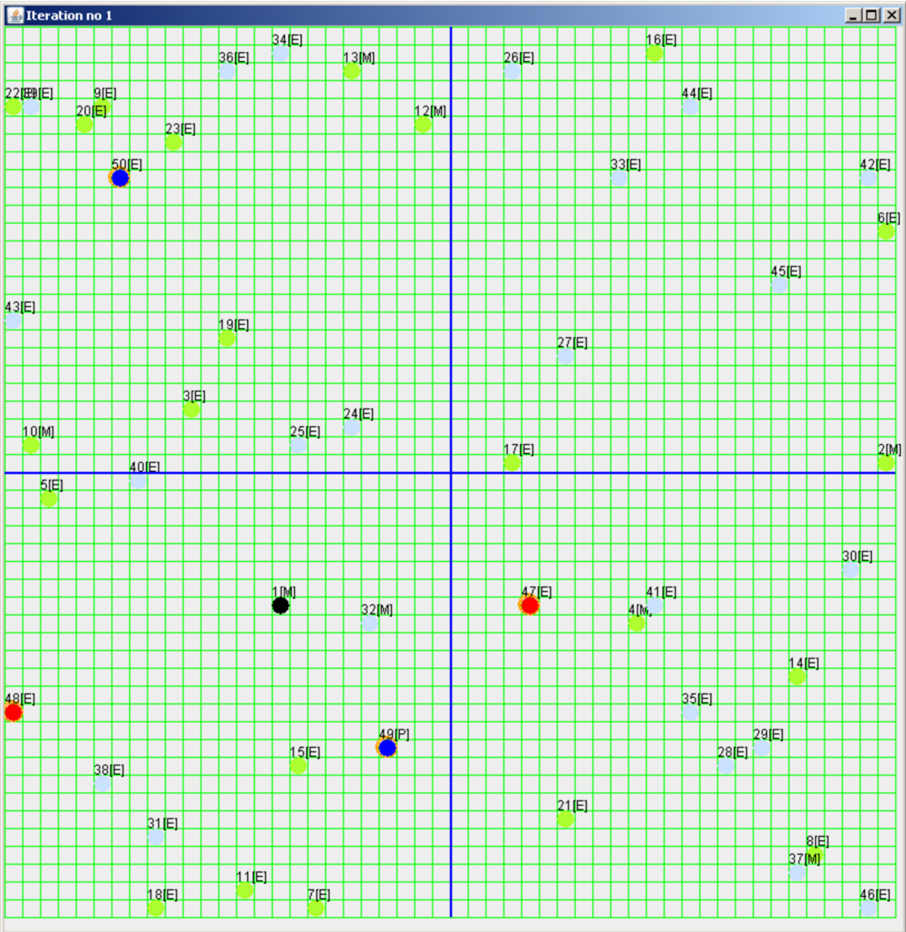


Fig. 7 Simulation snapshot of the set-up for studying dynamic norm change with 50 agents (littering agents in light blue, non-littering agents in light green, punishers of littering action in red and the punishers of eating action in blue)

after the punishers have left. In the absence of the punishers these agent may remove the norm from their identified norm set. A punisher that punishes eating actions enters this society at iteration 944. Hence some agents start inferring this norm. Note that the new agents enter an agent society with a clean slate (i.e. they do not have any norms). That is the reason for the fluctuations of both the lines in Figure 8. If there are more new agents in the society that are without norms, the percentage of the agents with the norm will be small.

Figure 9 shows how norms change in society 4. At the start of the simulation there is only one punisher in society 4, the punisher that punishes littering actions. So only one type is norm is inferred by the agents in the society. About iteration 164 this agent moves to society 2. Since there are no other agents to enforce the norm in the society, all the agents in this society gradually lose the norm by iteration

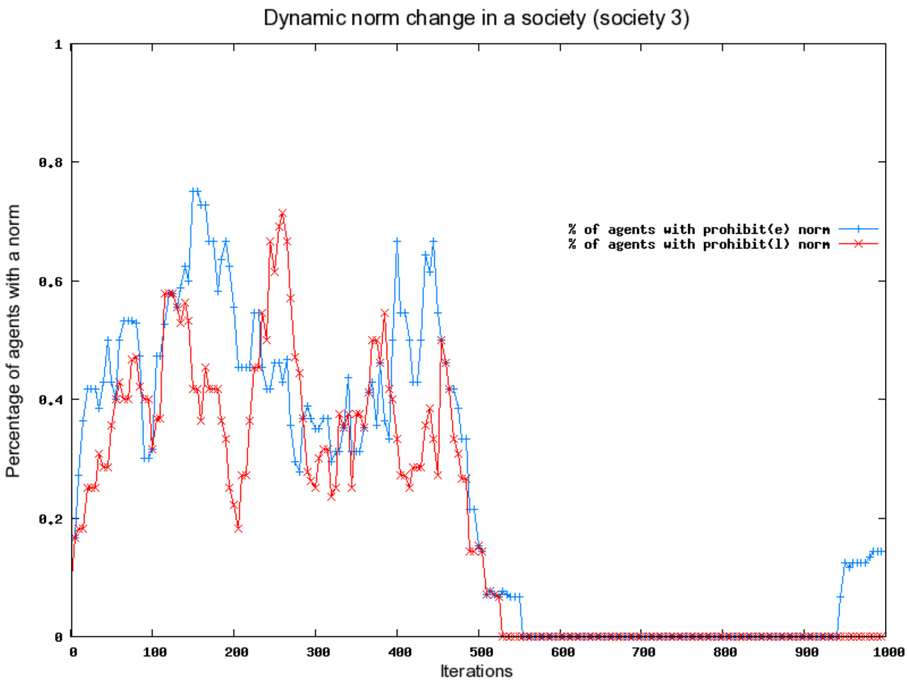


Fig. 8 Identification of two different norms (norm against eating, norm against littering) over 1000 iteration in society 3

200 (for the same reasons presented in the previous paragraph). About iteration 440, a punisher that punishes eating action moves to this society, so agents start inferring the norm against eating in the park, and when the punisher that punishes littering enters in iteration 463, the agents start inferring the norm against littering. About iteration 500, another punisher that punishes the littering action enters society 4 from society 2. Note that at this point there are more punishers that punish littering actions than eating actions. This is a reason for more agents inferring the norm against littering than eating. By iteration 944, the punisher that punishes the eating action moves to society 3. Hence the norm against eating starts decreasing in the society.

In these experiments (Figures 8 and 9) we have not specified what the criteria for emergence is. We have only shown the percentage of agents that have identified the existence of a norm based on a sanction. Researchers have used different criteria for norm emergence in a society (35% to 100% see [65]).

9.3 Experiment 3 - Dynamic norm change (from individual agent's view point)

An agent that moves across different societies may infer different type of norms. This experiment demonstrates that an agent is capable of inferring different norms in different societies using the norm identification architecture. Using the experimental set up described in section 9.2, we show how an agent can infer a norm (please see

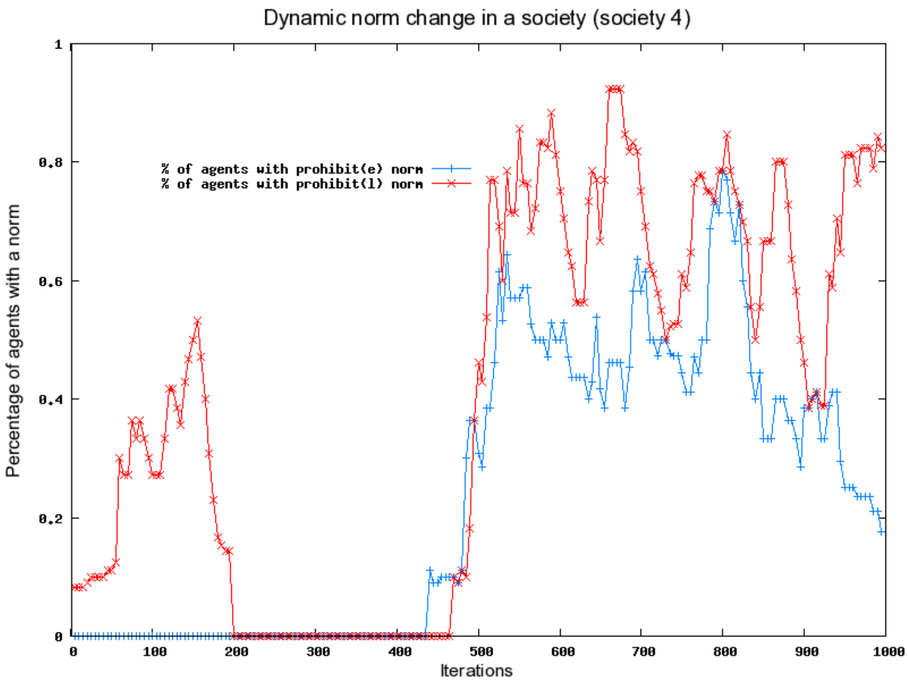


Fig. 9 Identification of two different norms (norm against eating, norm against littering) over 1000 iteration in society 4

the video referred to in Section 9.2). In the video, it can be observed that there is an agent in black. This black agent is the observer that moves between societies 3 and 4. Using our architecture, this agent is able to infer different types of norms (a norm against eating, a norm against littering, and a norm against both eating and littering). Figure 10 shows the result of norm inference for an agent at different points of time. It can be observed that an agent is able to infer the change of norm when it moves from one society to another. At different points of time, the agent will have different norms.

From the simulation video, it can be observed that the agent in black is initially in society 3. It infers the norm against eating in that society (shown as a blue region in Figure 10). It moves to society 4 in iteration 64. As society 4 has a littering punisher, it infers that norm in iteration 115 (shown as a red region in Figure 10). The agent then moves into society 3 in iteration 177, where both type of punishers are present. It then identifies the norm against littering first and then the norm against eating[24]. Note that it identifies both types of norms (co-existing norms[25]) in this society (regions with green diamonds in Figure 10) from iterations 250 to 268. As the agent moves in and

[24] This was because the probability of a punishment for littering was higher than eating. In the opposite case, the norm against littering would have been found first.

[25] In this work we have focused on simple co-existing norms (e.g. norms against eating and littering).

out of societies 3 and 4, it identifies different norms depending upon the punisher(s) present in the society.

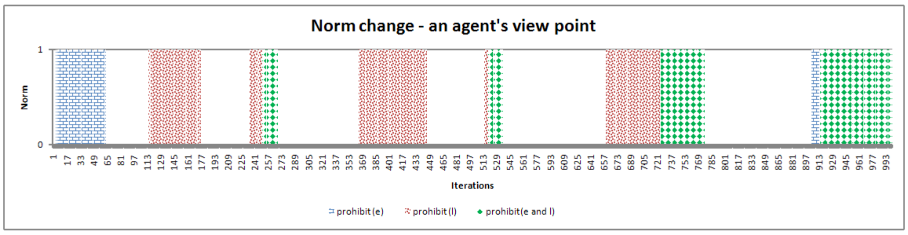


Fig. 10 Norm change identified by an agent over 1000 iterations - identification of individual norms (norm against eating in blue, norm against littering in red) and co-existing norms (norm against eating and littering together) in green

9.4 Experiment 4 - Adaptability of an agent

Norm Identification Frequency (NIF) and Norm Identification Threshold (NIT) are two parameters that an agent can vary based on it its success in recognizing a norm in the society. The Norm Identification Frequency (NIF) refers to how often an agent invokes the norm inference component to obtain a candidate norm set. For example if the NIF is set to five, an agent invokes the norm inference components once in five iterations. The Norm Identification Threshold (NIT) refers to the threshold that an agent sets in order to obtain a candidate norm set. For example if NIT is set to 50% for an agent, the agent will seek a candidate norm set which contains those sub-episodes whose occurrence probability is greater than or equal to 50%.

9.4.1 Adapting Norm Inference Frequency (NIF)

An agent in our set-up starts with NIF=5. When it does not find any norm it retains the same NIF value. Once it has found all the norms in the society (i.e. there are no new norms to be found after a certain number of iterations), it increases its NIF value by a discrete amount (an increase of 5 in our case[26]). The new value of NIF will be 10 which implies that the agent will infer norms only once in 10 iterations. This continues till a norm changes (e.g. a new norm is found). When a new norm is found, the NIF is set to back to 5. Figure 11 shows how the NIF changes by keeping NIT constant (50% in our case). It should be noted that when an agent moves into a new society, the NIF is set to 5.

Figure 11 shows two lines corresponding to an agent moving in one society and within four societies with static punishers. All these societies have only one type of

[26] We have used a value of 5 in our simulations to model small fluctuations (increase or decrease) in the NIF value. This discrete amount is a parameter in our system which can be changed.

punisher (i.e. only one type of norm can be inferred). It can be observed that when the agent moves within one society, it infers the norm, and hence its NIF value increases (as the norm does not change). In the case of the agent moving across four different societies, the agent's NIF increases after it has found a norm, as long as it is in the same society where it found the norm. When the agent moves to a new society[27], the agent's NIF is set to the base value, and then it starts increasing once it has found the norm again. The "sudden jumps in values" (i.e. the vertical lines) that occur in regular intervals indicate that an agent has moved from one society to another.

It should be noted that when the punishers are moving between societies (not shown in the figure), the NIF values (i.e. the NIF line) of an agent is different from when the punishers are static. In a set-up with static punishers, the punishers do not move (i.e. they remain in one society), but the other agents move. In a set-up with moving punishers, both the punishers and the other agents can move across societies. Hence, the NIF values for the same agent in these two set-ups are different. For example, let us assume that there is only one punisher in society 1. When the punisher moves from society 1 to 2, agents in society 1 are not able to recognize the norm when they try to infer the norm the next time (as dictated by their NIF value) as there are no sanctions. So, they may decrease their NIF value. On the other hand, in the set-up where punishers do not move, the agents in society 1 would have inferred the norm which would result in an increase in their NIF value.

9.4.2 Adapting Norm Inference Threshold (NIT)

An agent in our set-up starts with NIT=50%. When an agent invokes the norm inference component (NIF value is met), it initially uses the default NIT value. When the norms do not change the agent increases the NIT value by a discrete amount (an increase of 5 in our case). When no norm is found, it decreases the NIT value by a discrete amount (a decrease of 5 in our case). An agent increases its NIT because it can reduce extra computation that is needed to infer a candidate norm. For example if the same set of norms are obtained when an agent has NIT=100%, there is no reason why an agent should retain the value of NIT=50% where it has to perform some additional computation to infer the same norm (based on the PNI algorithm). An agent reduces the NIT in order to explore the search space of candidate norms below its initial threshold value. Note that when an agent moves into a new society, the NIT is set to 50%. Figure 12 shows how the NIT changes by keeping NIF constant (NIF=5 in our case). The vertical lines that occur periodically indicate that an agent has moved from one society to another. The initial value of NIT is set to 50%, because an agent should not have to start from scratch to infer a candidate norm (i.e. from NIT=0%) in order to avoid extra computation.

It can be noted from Figure 12 that when an agent moves within one society with one type of norm, its NIT threshold gradually reaches the value of 100, as it is able to infer the norms with a high level of support. In the case of the agent moving across societies, an agent's NIT value starts increasing when the agent has identified the

[27] We assume that the agent knows when it enters a new society. For example, the agent may know the physical boundaries of the society in which it is situated. For example, an avatar in SecondLife explicitly knows that it is moving from the location of a community to another.

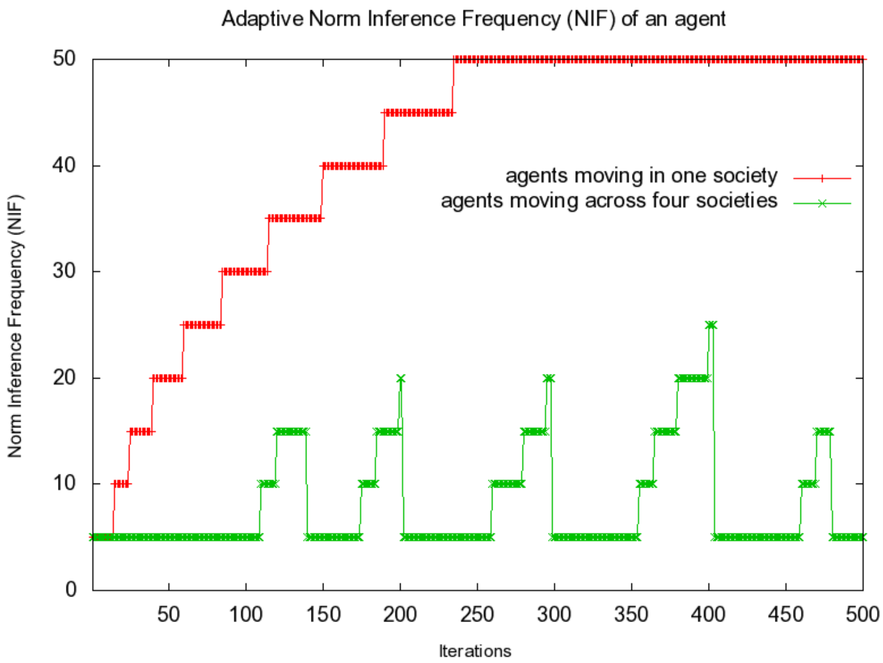


Fig. 11 Comparison of an agent's ability to adapt Norm Inference Frequency (NIF) when it moves within one society vs. when it moves in four different societies (over 500 iterations)

norm, but it drops to 50 when it moves from one society to another. For the same set-up when the punishers are moving, the agent may not be able to infer the norms when the punishers move into a new zone. So, the NIT line showing the movement of an agent with static punishers is different from the scenario involving dynamic punishers (not shown in the figure).

9.4.3 Adapting both NIF and NIT

An agent can vary both NIF and NIT. Figure 13 shows how both of these variables change in an agent. The circled region is of interest, because it shows that an agent did not infer a norm initially (iteration 80 to 111). So the NIT line drops. It inferred the norm around iterations 111 to 133, which is indicated by the upwards trend. Then the agent did not infer the norm between iterations 134 to 139. Hence there is a drop in the NIT line. In iteration 140, the agent has moved to a new zone. The NIF line is similar to the one shown in Figure 11. The line that appears in the bottom shows when an agent moves from one society to another.

Experiments reported in sections 9.2, 9.3 and 9.4 have demonstrated that an agent can dynamically change its norm inference behaviour. An agent, being an autonomous entity, can decide when to increase or decrease these parameters to infer norms.

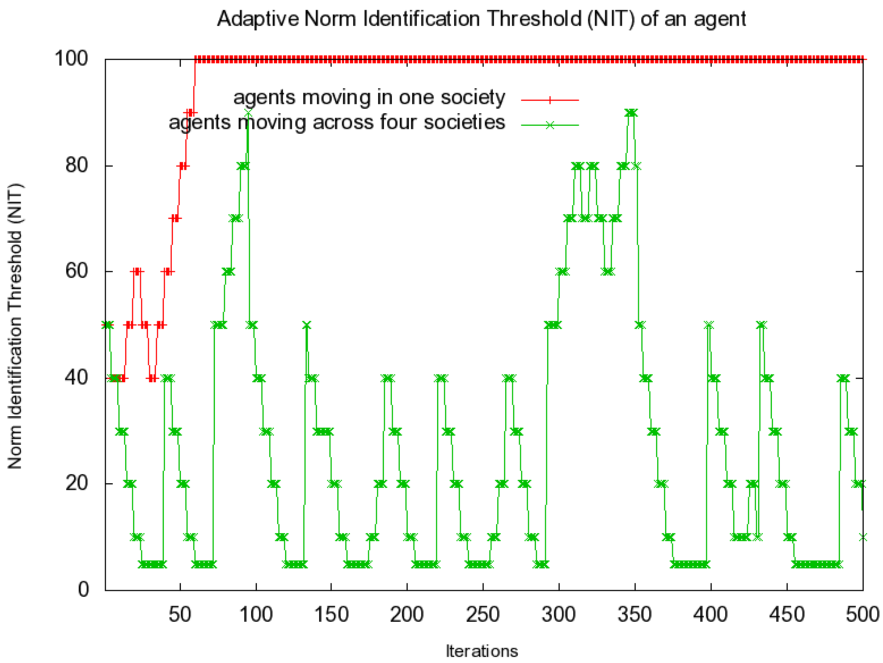


Fig. 12 Comparison of an agent's ability to adapt Norm Inference Threshold (NIT) when it moves within one society vs. when it moves in four different societies (over 500 iterations)

9.5 Experiment 5 - Using norm history

When an agent moves from one society to another, it can record the norm of the society it is leaving, so that the information can be used when it comes back to the same society. The experimental results shown in Figures 14 (a) and (b) demonstrate that the percentage of agents that have identified a particular norm in an agent society is high when agents store the norm (possess history) of the society. The recorded history can be used when an agent re-enters the society that it has previously been to. The graphs given in Figures 14 (a) and (b) have two lines each. The serrated red line shows the percentage of agents in a society with or without a norm. The straight line in green shows the average proportion of agents in the society with or without a norm.

Note that researchers have studied the impact of amount of history recorded by agents on convention emergence [68]). The objective of this experiment is to compare how much better the agent will be if it records the norms of the society that it leaves. It can be seen that on average 77% of the agents can infer norms when history was stored (Figure 14 (b)) while only 47% of the other agents on average inferred norms (Figure 14 (a)). The experimental set-up was the same for both these experiments. A higher percentage of agents inferred norms when using their history, because the agents that come in with history information can start asking other agents in the society whether a norm holds (norm verification). If an agent does not have a norm

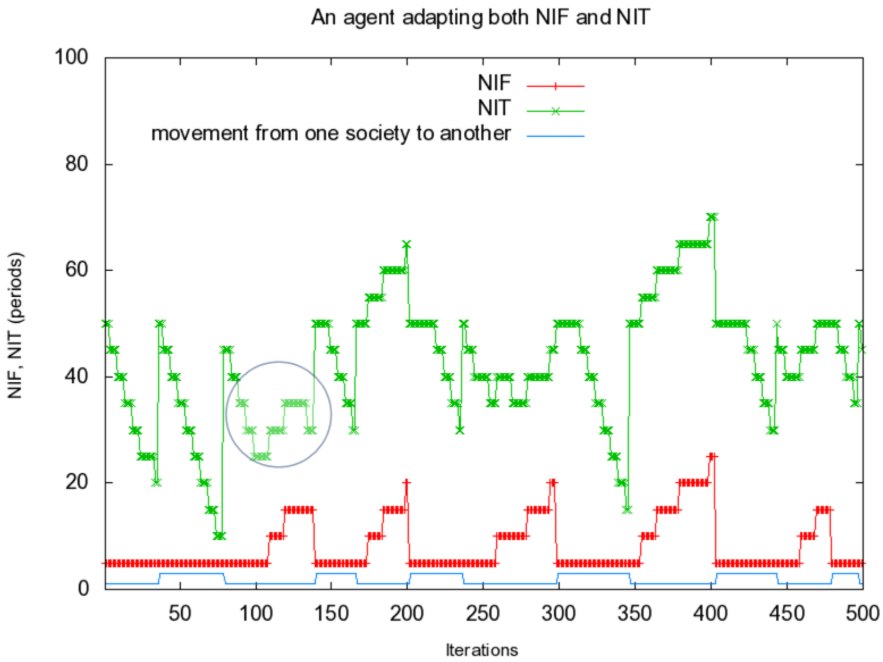


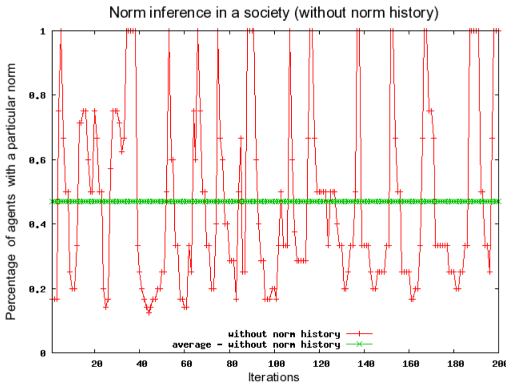
Fig. 13 Comparing the ability of an agent to adapt both Norm Inference Frequency (NIF) and Norm Inference Threshold (NIT) over 500 iterations when it moves across societies

history, it first has to infer the norm (i.e. invoke the norm inference component) and then ask another agent for norm verification, which is slower than using the norms in the norm history at the verification stage.

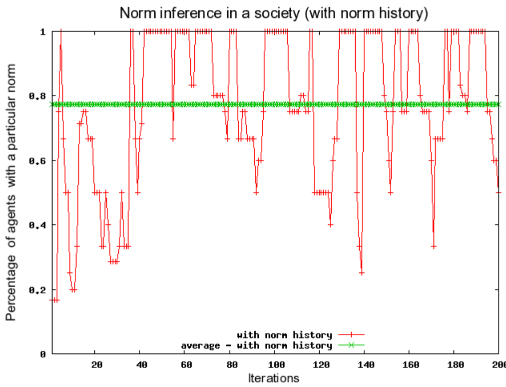
Using history is useful when the punishers are not moving (i.e. the norms in the society are stable). If the punishers are moving, then the norms may change (i.e. when there are different types of punishers). If the norms change, then the mechanism may not be very useful. If there are a large number of separate societies, then the agent may not come back to a previously inhabited society. In this case, the history information may not be useful. However, the agent is better off keeping the history if it comes back to a society it has previously been to, since it does not have to start inferring the norms from scratch.

9.6 Experiments based on the utility of an agent

An agent, being an autonomous entity, may choose to exercise its autonomy in order to maximize its utility. Such utilitarian agents may choose to become a part of a society that leads to an an optimization of their utility. In this section we describe two experiments that we have conducted using utilitarian agents. The objectives of experiments are two-fold.



(a) Without norm history



(b) With norm history

Fig. 14 Comparison of the percentage of agents in the society that have identified norms (with and without making use of norm history) over 200 iterations

1. To demonstrate that the utility of a norm-abiding agent is better in a normative society where punishers are present than in a society where they are absent (i.e. a society where norm violations are not punished).
2. To demonstrate that when agents are capable of norm inference, the norm establishment in a society is faster than when agents do not infer norms.

9.6.1 Experimental set-up

Let us assume that there are two societies: a normative society and a society with no norms. There are three types of agents: learning litterers (LL), non-litterers (NL) and non-littering punishers (NLP) in both the societies. An agent has a utility value which we call the satisfaction level (S) which varies from 0 to 100.

An agent's satisfaction level (S) decreases in the following situations:

– When a non-litterer observes a littering action its satisfaction level decreases (-1).

– When a litterer is punished, its utility decreases (-1).
– For all agents, littering activity results in the decrease of the utility. This is because each littering activity ruins the *commons* area (-1/number of agents in the society)[28].

An agent's satisfaction level (S) increases (i.e. it gains utility) in the following situations:

– When a litterer litters, it gains utility in a society (+1).
– When a non-litterer does not see any littering action in a society, its utility increases (+1).

At the start of experiments, an agent moves across two societies (society 1 and 2). All the agents start with a satisfaction level of 50. This value can increase or decrease based on the society the agent is in. When the agents move into a new society, the satisfaction level is set to 50. When an agent has been to both the societies, an agent chooses the society for which it has a higher utility.

9.6.2 Experiment 1 - Norm abiding agents are better off in a normative society

Using the experimental set-up described in the previous section, we experimented how the agents organized themselves into two societies based on utility. In one of the societies, the punishers were present (society 2). In the other society the punishers were absent. In this experiment, the agents do not make use of the norm inference mechanism. They are utility maximizers (i.e. they would move to a society that yields better utility).

At the start of the experiment, the agents moved in both societies. At regular intervals of time (every 50 iterations) each agent decides which society to choose. The agent evaluates its satisfaction level based on the societies it inhabited. The litterers' utility in society 1 is better than in society 2, because in society 2 they are punished. So they move towards society 1. As the litterers move towards society 1, the non-litterers move towards society 2, because their utility in that society is better, due to the absence of litterers. It can be observed in Figure 15 that at the end 160 iterations, the litterers have moved from society 2 to society 1. When there are litterers in society 1, the non-litterers move to society 2. It can be observed that at the end of 350 iterations all the non-litterers have moved into society 2.

We also conducted experiments by making the punishers move across societies at certain intervals of time. In this set-up, we ran the experiments for 1500 iterations[29]. In the first 500 iterations, the punishers were in society 2, and in the second 500 iterations the punishers moved to society 1. In the third 500 iterations the punishers moved back to society 2.

---

28 Cleaning *commons* area such as a public park that has been littered would be at a cost which would be paid by the rate-payers of the city. From this point of view, bearing the cost of a litter has been distributed to all members of the society including the litterer.

29 Simulation can be viewed at http://unitube.otago.ac.nz/view?m=SE7M11esZnI. The litterers are in black, non-litterers are in green and the punishers are in red.

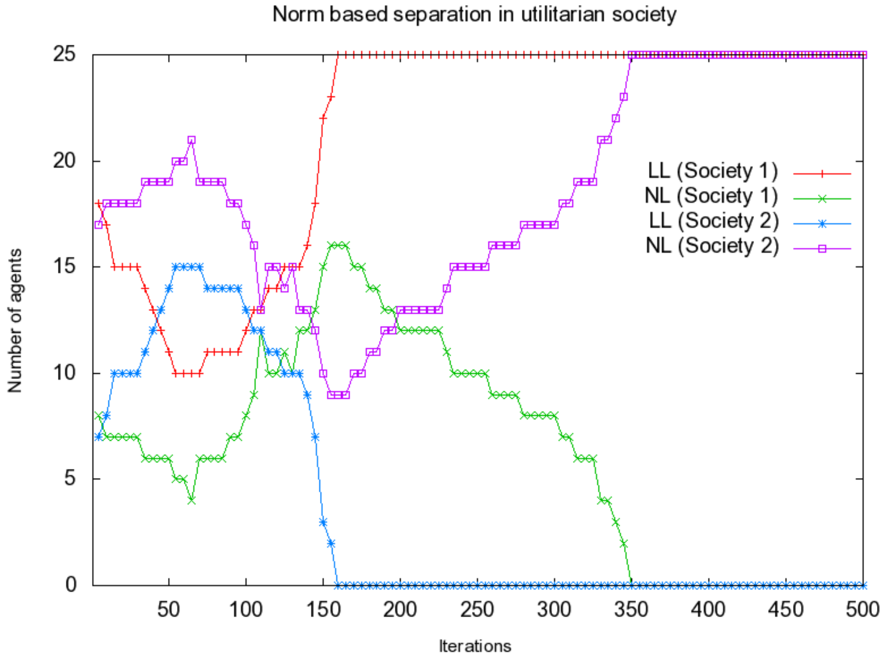


Fig. 15 Separation of agents into two groups based on utility over 500 iterations - litterers move to a society where there are no punishers (society 1) and non litterers move to a society where punishers are present (society 2)

In the first 500 iterations, the agents were separated into two groups, as described above. After 500 iterations, when the punishers move to society 1, the utility of litterers in that society starts decreasing, so the litterers move to society 2. When they move to society 2, the non-litterers' utility decreases, so they move to society 1. Again it was observed that the two societies were separated based on the personality of the agents (society 1 with non-litterers and society 2 with litterers). After 1000 iterations when the punishers have moved to society 2, the same process continues. Society 1 now has all the litterers and society 2 has non-litterers.

This experiment demonstrates that norm-abiding agents are better off in the normative society, where the norm violation is punished by sanctioning agents, and the non-litterers are better off to be in a society with no norms. This experiment also demonstrates that a normative agent is adaptive, as it moves from one society to another if its utility decreases in the society.

9.6.3 Experiment 2 - Utilitarian vs. Hybrid strategies for agents

In the previous experiment the agents employed the utilitarian strategy. In this experiment the agents use the norm inference mechanism. At the same time they also compute utility. We call this a hybrid strategy. Except for the change of the strategy the experimental set up was similar to the previous experiment. We observed that the overall separation of the two groups is faster when the agents are able to apply the norm-inference mechanism along with the utilitarian mechanism. This is because when the litterers in society 2 infer that there is a norm against littering, then they will decide not to litter in the society (to minimize the decrease in their utility) and also decide to move to another society. When a non-litterer in society 2 infers that there is a norm then it decides to stay in that society, as it knows that it would be better off in this society (as the punishers will punish the littering agents). So, in this experiment the separation of agents into two societies is faster when a norm inference mechanism is used along with the utilitarian mechanism.

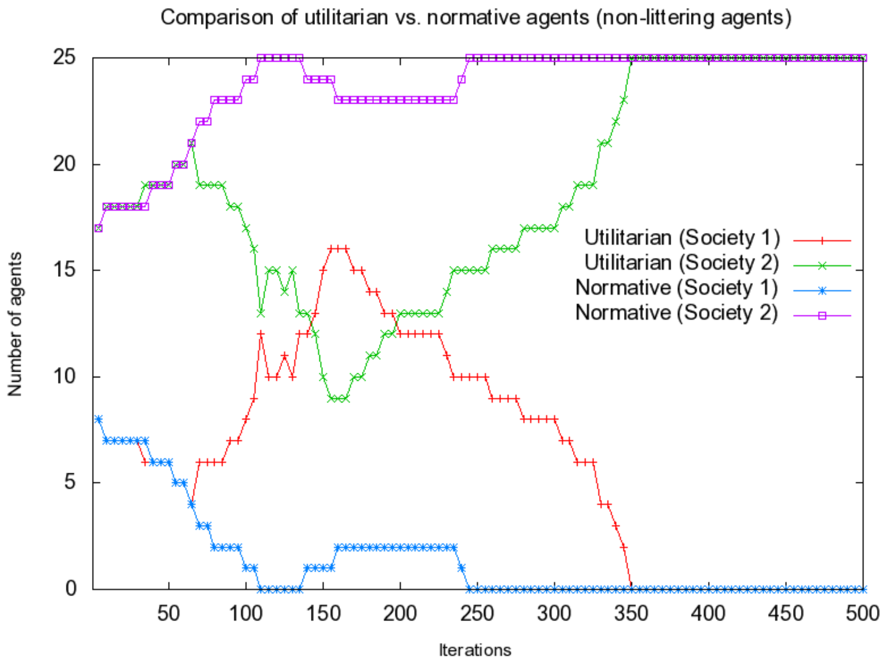


Fig. 16 Comparison of utilitarian vs. hybrid (normative-utilitarian) strategies - separation of agents into two groups (norm followers and norm violators) over 500 iterations

Figure 16 shows the number of litterers and non-litterers in two societies with the two types of strategies for the non-littering agents. It can be noted that when the hybrid strategy was employed by the agents, the separation of agents into two separate groups was faster than when the agents used just the utilitarian strategy. The hybrid strategy resulted in the littering agent moving from society 2 to society 1 faster than when using utilitarian society. In this experiment, the system using hybrid strategy converged 100 iterations earlier than the system that used utilitarian strategy. As the litterers moved to society 1, the non-litterers moved to society 2.

10 Discussion

In this section we first make some remarks on the scalability of the system. Second, we describe the main contributions of this paper, its limitations and the future work.

10.1 Scalability

We have conducted scalability experiments by varying the number of domain-specific unique events considered in the system to 8, 16 and 24[30]. We note that the results obtained across different numbers of events were consistent (e.g. decreasing NIT increases the chances of norm identification). It should also be noted that when an agent increases these parameters the amount of computation required increases. An agent can set a maximum limit to which it increases these parameters in order to avoid computation costs which can be guided at the design time of the agent, based on the domain considered. Another approach to handle high computation costs in certain domains is to offload the computational process to distributed servers (not adopted in our simulation system). Further, servers can employ complex event processing engines such as Esper [35] to execute algorithms developed in this paper to extract norms.

10.2 Main contributions

The main contributions of this paper are the following:

1. The question of “how an agent comes to find out what the norms of society are” has not received much attention in the field of normative multi-agent systems. We have made progress in that regard by proposing a norm identification architecture from the perspective of a single agent (i.e. internal agent architecture for norm identification). Most researchers agree that there will be some form of sanction for norm violation once a norm is established (e.g. [23, 33]). Hence, the notion of a sanction has been considered to be a top level entity in our work. We have assumed that even when a norm is being created, the notion of sanction is important for norm identification.
2. The proposed architecture uses Prohibition Norm Identification (PNI) algorithm to identify potential norms. The PNI algorithm makes use of a modified version of the association rule mining algorithm called WINEPI. To our knowledge this is the first work that makes use of a association rule mining approach for norm identification. In this context we have demonstrated the following:
   – An adaptive agent can infer norms based on its success in identifying norms (by changing different parameters of the system if it is not successful). For example, an agent can modify the norm inference threshold and norm inference frequency parameters to infer potential norms.

[30] We note this is an improvement from the game-theory based works which have typically considered only two options for agents based on coordination or cooperation games [60, 67, 68, 79] (e.g. drive on the left or the right). Additionally those works do not deal with sanctions, therefore only cater for recognizing a convention.

  - An agent can dynamically add, remove and modify norms. An agent in our framework can add an identified norm to its norm set and also remove the norm when it does not hold.
  - An agent can identify co-existing norms. Agents employing our framework can infer co-existing norms in a park such as a norm against littering and a norm against eating.
3. Through simulations we have shown how the architecture allows for detection (i.e. identification), communication (i.e. verification) and modification (i.e. dynamic change of norms) of norms. We have also demonstrated that the norm inference mechanism is beneficial for an agent as it learns about a norm faster than just using a utilitarian strategy.

We believe this architecture can be used in several settings. For example, the norm identification architecture can be used to infer norms in Massively Multi-player Online Role Playing Games (MMORPGs). Players involved in massively multi-player games perform actions in an environment to achieve a goal. They may play as individuals or in groups. When playing a cooperation game (e.g. players forming groups to slay a dragon), individual players may be able to observe prescriptions or proscriptions of actions that are allowed within the group. The normative architecture proposed in this paper can be used to identify norms that are being formed. Secondly, the mechanisms presented in the paper can be used as a starting point to infer norms from virtual environments such as Second Life[31]. Thirdly, software agents engaging in e-commerce activities such as buying and selling goods, can infer norms using this architecture. We note that the same architecture can be used for identifying the violation of obligations. In this case, instead of looking for sequence of actions that could have caused a sanction, an agent can look for the absence of an event or a set of events that could have caused a sanction[32]. In the e-commerce scenario it could be the absence of a *pay* action within certain period of time that could be a reason for a sanction.

10.3 Limitations and future work

- **Only observable events are identified as norms** - In our work, only observable events can be identified as norms (e.g. littering a park). Consider a scenario where someone might receive a sanction two days after they had done something wrong (e.g. receiving a speeding ticket after two days). In this case, the event that caused the sanction may not necessarily be observed since an agent may not store a very long history. In those cases, the present approach will not work[33]. Our work will suit social norms where the sanction is immediate for the

[31] We acknowledge that the scenarios that can be modeled using Second Life are far richer than the ones presented in this paper. Those scenarios will require richer norm identification mechanisms.

[32] This has been demonstrated elsewhere. In the case of identifying both types of norms together, first obligation norms should be inferred followed by the prohibition norms in order to avoid false positives of prohibition norms (future work).

[33] This is a granularity issue. An agent can choose to record all the events or only some of the key events that happen around it. By choosing to record key events, the agent may have a coarser but longer

events that are observable. We believe most social norms have immediate sanctions. However, sanctioning mechanisms such as the lowering of the reputation may not be directly observable as the action is internal to an agent. Our approach works for sanctioning mechanisms that have explicitly visible and interpretable actions (e.g. gestures such as shaking head in disapproval and yelling or visible chat messages).

– **Consideration of simple roles for agents** - In this work, we have defined simple roles for agents. For example, the park scenario the agents had three roles: litterer, punisher and non-litterer. We believe our work can be extended to include multiple roles for the same agent and also include hierarchies of roles. In the future richer and complex domains can be considered.
– **Alternative mechanisms to norm verification** - An agent can verify whether its candidate norms hold by undertaking actions that it observes to be sanctioned (e.g. by violating the tipping norm). Based on the outcome of tests the agent carries out, it could infer what the norms could be. This would be a meta-level norm-testing mechanism of an agent. This would also solve the lying problem (i.e. a norm verifying agent may potentially lie about whether a norm holds in a particular society).
  Another alternative to norm verification is to include humans in the loop. The agents that are proxies to humans can recommend candidate norms to humans and the humans can then decide which ones are norms. Agents in this case play the role of a norm recommender. Such a norm recommender system can recommend both the norm (e.g *prohibit(litter)* might be the norm which is inferred based on an agent identifying the fact that whenever there is a sanction, littering action is involved 100% of the time) and the norm strength (e.g. the frequency of punishing the littering activity is 50%). These extended mechanisms can be explored in the future.
– **Emergence of signals (sanctions)** - Our current work assumes that a punisher agent knows a priori the action that should be sanctioned. Though this may hold for norm leader or entrepreneur agents that come up with norms, in some scenarios, the action that is sanctioned may not be known even to the potential sanctioning agent ahead of time. The sanction might emerge depending upon the environmental dynamics. For example, an agent might not sanction if it sees one agent littering. But, when it sees *n* agents littering the park, it might start punishing, because that action has lowered its utility beyond a certain threshold (an internal utility function). In this scenario, an agent can use a learning algorithm (e.g. Q-Learning [81]) to identify an action that lowers its utility and then can sanction that action. The norm then can be identified from the sanction that has emerged. The extension here would be to include a mechanism for the emergence of sanctions and then apply the mechanisms proposed in this thesis to identify the norm.

history which can be used to infer norms. Additionally, if sanction follows long after the action has been performed, it is difficult to associate a sanction with the action that triggered it (i.e. it is computationally expensive).

11 Conclusion

The question *how an agent comes to find out what the norms of society are* has not received much attention in the field of normative multi-agent systems. To that end, we have proposed a norm identification architecture from the perspective of a single agent (i.e. an internal agent architecture for norm identification). Many researchers agree that there will be some form of sanction once a norm is established (e.g. [23, 33]). Hence, the notion of a sanction has been established as a top level entity in our work. We have assumed that even when a norm is being created, the notion of a sanction is important for norm identification. This paper describes how an agent can infer prohibition norms in a society using the norm identification architecture. The Prohibition Norm Inference (PNI) algorithm presented in this paper can be used to identify potential prohibition norms. The proposed PNI algorithm makes use of a modified version of the association rule mining algorithm. We have shown that an adaptive agent can infer norms by varying different attributes of the system. Through simulations we have shown how the norm identification architecture allows for detection (i.e. identification), communication (i.e. verification) and modification (i.e. dynamic change of norms) of norms. We have also demonstrated that the norm inference mechanism is beneficial for an agent, as it learns about a norm faster than just using a utilitarian strategy.

12 Appendix

Acronyms and expansions of the terms used in this paper are given below based on the alphabetical ordering of the acronyms.

Table 3 Acronyms used and the corresponding expansions

Acronym	**Expansion**
CPNS	Candidate Prohibition Norm Set
ES	Event Sequences
HL	History Length
NIT	Norm Identification Threshold
NIF	Norm Inference Frequency
OP	Occurrence Probability
SEE	Special Event Episode
SEEL	Special Event Episode List
SEL	Sub episode list
SES	Special Event Set
UES	Unique Event Set
WS	Window Size

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: J.B. Bocca, M. Jarke, C. Zaniolo (eds.) Proceedings of 20th International Conference on Very Large Data

Bases (VLDB'94), Santiago de Chile, Chile, pp. 487–499. Morgan Kaufmann (1994)
2. Akerlof, G.A.: The economics of caste and of the rat race and other woeful tales. The Quarterly Journal of Economics **90**(4), 599–617 (1976)
3. Aldewereld, H., Dignum, F., García-Camino, A., Noriega, P., Rodríguez-Aguilar, J.A., Sierra, C.: Operationalisation of norms for usage in electronic institutions. In: Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp. 223–225. ACM Press, New York,NY, USA (2006)
4. Andrighetto, G., Campenni, M., Cecconi, F., Conte, R.: The complex loop of norm emergence: A simulation model. In: S.H. Chen, C. Cioffi-Revilla, N. Gilbert, H. Kita, T. Terano, K. Takadama, C. Cioffi-Revilla, G. Deffuant (eds.) Simulating Interacting Agents and Social Phenomena, *Agent-Based Social Systems*, vol. 7, pp. 19–35. Springer (2010)
5. Andrighetto, G., Conte, R., Turrini, P., Paolucci, M.: Emergence in the loop: Simulating the two way dynamics of norm innovation. In: G. Boella, L. van der Torre, H. Verhagen (eds.) Normative Multi-agent Systems, no. 07122 in Dagstuhl Seminar Proceedings. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany (2007)
6. Arcos, J.L., Esteva, M., Noriega, P., Rodríguez-Aguilar, J.A., Sierra, C.: Environment engineering for multiagent systems. Engineering Applications of Artificial Intelligence **18**(2), 191204 (2005)
7. Axelrod, R.: An evolutionary approach to norms. The American Political Science Review **80**(4), 1095–1111 (1986)
8. Bandura, A.: Social Learning Theory. General Learning Press (1977)
9. Becker, G.S.: The Economic Approach to Human Behavior. University of Chicago Press (1978)
10. Bench-Capon, T., Coenen, F., Leng, P.: An experiment in discovering association rules in the legal domain. In: 11th International Workshop on Database and Expert Systems Applications, 2000, pp. 1056 –1060 (2000)
11. Bicchieri, C.: The Grammar of Society: The Nature and Dynamics of Social Norms. Cambridge University Press, New York, USA (2006)
12. Boella, G., van der Torre, L.: The ontological properties of social roles in multi-agent systems: definitional dependence, powers and roles playing roles. Artificial Intelligence and Law **15**(3), 201–221 (2007)
13. Boella, G., van der Torre, L.: An architecture of a normative system: counts-as conditionals, obligations and permissions. In: Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp. 229–231. ACM Press, New York, NY, USA (2006)
14. Boella, G., van der Torre, L., Verhagen, H.: Introduction to normative multiagent systems. Computational and Mathematical Organization Theory **12**(2-3), 71–79 (2006)
15. Boella, G., van der Torre, L., Verhagen, H.: Introduction to the special issue on normative multiagent systems. Autonomous Agents and Multi-Agent Systems **17**(1), 1–10 (2008)
16. Boman, M.: Norms in artificial decision making. Artificial Intelligence and Law **7**(1), 17–35 (1999)
17. Boyd, S., Ghosh, A., Prabhakar, B., Shah, D.: Randomized gossip algorithms. IEEE Transactions on Information Theory **52**(6), 2508–2530 (2006)
18. Broersen, J., Dastani, M., Hulstijn, J., Huang, Z., van der Torre, L.: The boid architecture: conflicts between beliefs, obligations, intentions and desires. In: AGENTS '01: Proceedings of the fifth international conference on Autonomous agents, pp. 9–16. ACM, New York, NY, USA (2001)
19. Campbell, E.Q.: The internalization of moral norms. Sociometry **27**(4), pp. 391–412 (1964)
20. Candale, T., Sen, S.: Effect of referrals on convergence to satisficing distributions. In: Proceedings of the fourth international joint conference on Autonomous Agents and MultiAgent Systems, (AAMAS'05), pp. 347–354. ACM, New York, NY, USA (2005)
21. Ceglar, A., Roddick, J.F.: Association mining. ACM Computing Surveys **38**(2), 5 (2006)
22. Chwe, M.S.Y.: Rational Ritual: Culture, Coordination and Common Knowledge. Princeton University Press (2001)
23. Coleman, J.: Foundations of Social Theory. Belknap Press (1990)
24. Conte, R., Castelfranchi, C.: Understanding the effects of norms in social groups through simulation. In: N. Gilbert, R. Conte (eds.) Artificial societies: the computer simulation of social life, pp. 252–267. UCL Press, London (1995)
25. Conte, R., Castelfranchi, C.: From conventions to prescriptions. Towards an integrated view of norms. Artificial Intelligence and Law **7**(4), 323–340 (1999)
26. Conte, R., Dignum, F.: From social monitoring to normative influence. Journal of Artificial Societies and Social Simulation **4**(2) (2001)
27. Conte, R., Paolucci, M.: Intelligent social learning. Journal of Artificial Societies and Social Simulation **4**(1) (2001)

28. Cranefield, S., Li, G.: Monitoring social expectations in second life. In: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 2, AAMAS '09, pp. 1303–1304. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC (2009)
29. Criado, N., Argente, E., Botti, V.: Rational strategies for autonomous norm adoption. In: Proceeding of the Ninth international workshop on Coordination, Organization, Institutions and Norms in agent systems (COIN@AAMAS 2010), pp. 9–16 (2010)
30. Dictionary.com: empirical. http://dictionary.reference.com/browse/empirical (2010). Accessed 12 November 2010
31. Dignum, F.: Autonomous agents with norms. Artificial Intelligence and Law **7**(1), 69–79 (1999)
32. Dignum, F., Kinny, D., Sonenberg, L.: From desires, obligations and norms to goals. Cognitive Science Quarterly **2**(3-4), 407–430 (2002)
33. Elster, J.: Social norms and economic theory. The Journal of Economic Perspectives **3**(4), 99–117 (1989)
34. Epstein, J.M.: Learning to be thoughtless: Social norms and individual computation. Computational Economics **18**(1), 9–24 (2001)
35. EsperTech: Esper tutorial. http://esper.codehaus.org/tutorials/tutorial/tutorial.html (2010). Accessed 15 June 2011
36. Gaertner, D., García-Camino, A., Noriega, P., Rodríguez-Aguilar, J.A., Vasconcelos, W.W.: Distributed norm management in regulated multi-agent systems. In: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems (AAMAS'07), pp. 624–631. ACM, Honolulu, Hawaii (2007)
37. García-Camino, A., Rodríguez-Aguilar, J.A., Sierra, C., Vasconcelos, W.W.: Norm-oriented programming of electronic institutions. In: Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp. 670–672. ACM Press, New York, NY, USA (2006)
38. Gibbs, J.P.: Norms: The problem of definition and classification. American Journal of Sociology **60**, 586–594 (1965)
39. Governatori, G., Stranieri, A.: Towards the application of association rules for defeasible rule discovery. In: Fourteenth annual international conference on Legal Knowledge and Information Systems, vol. 4, pp. 63–75. IOS Press (2001)
40. Habermas, J.: The Theory of Communicative Action, Volume 1: Reason and the Rationalization of Society. Beacon Press (1985)
41. Hamada, M.: Web-based environment for active computing learners. In: ICCSA (1), pp. 516–529 (2008)
42. Han, J., Kamber, M.: Data mining: concepts and techniques. Morgan Kaufmann (2006)
43. Hoffmann, M.J.: Entrepreneurs and Norm Dynamics: An Agent-Based Model of the Norm Life Cycle. Tech. rep., Department of Political Science and International Relations, University of Delaware, USA (2003)
44. Johansson, M., Verhagen, H.: Massively multiple online role playing games as normative multiagent systems. In: G. Boella, P. Noriega, G. Pigozzi, H. Verhagen (eds.) Normative Multi-Agent Systems, no. 09121 in Dagstuhl Seminar Proceedings. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, Dagstuhl, Germany (2009)
45. Jones, A.J.I., Sergot, M.J.: A formal characterisation of institutionalised power. Logic Journal of the IGPL **4**(3), 427–443 (1996)
46. Koda, T., Ruttkay, Z.: Cultural differences in using facial parts as cues to recognize emotions in avatars. In: IVA '09: Proceedings of the 9th International Conference on Intelligent Virtual Agents, pp. 517–518. Springer-Verlag, Berlin, Heidelberg (2009)
47. López y López, F.: Social Powers and Norms: Impact on Agent Behaviour. Ph.D. thesis, Department of Electronics and Computer Science, University of Southampton, United Kingdom (2003)
48. López y López, F., Luck, M., d'Inverno, M.: Constraining autonomy through norms. In: Proceedings of The First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), vol. 2, pp. 674–681. ACM, New York, NY, USA (2002). URL citeseer.ist.psu.edu/lopez02constraining.html
49. López y López, F., Márquez, A.A.: An architecture for autonomous normative agents. In: Proceedings of the Fifth Mexican International Conference in Computer Science (ENC), pp. 96–103. IEEE Computer Society, Los Alamitos, CA, USA (2004)

50. Lotzmann, U., Möhring, M.: Simulating norm formation: an operational approach. In: Proceedings of the Eighth International Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp. 1323–1324. International Foundation for Autonomous Agents and Multiagent Systems (2009)
51. Mannila, H., Toivonen, H., Inkeri Verkamo, A.: Discovery of frequent episodes in event sequences. Data Mining and Knowledge Discovery **1**(3), 259–289 (1997)
52. McAfee, R.P., McMillan, J.: Auctions and bidding. Journal of Economic Literature **25**(2), pp. 699–738 (1987)
53. Meneguzzi, F.R., Luck, M.: Norm-based behaviour modification in BDI agents. In: Proceedings of the Eighth International Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp. 177–184. IFAAMAS (2009)
54. Neumann, M.: A classification of normative architectures. In: K. Takadama, C. Cioffi-Revilla, G. Deffuant (eds.) Simulating Interacting Agents and Social Phenomena: The Second World Congress, *Agent-Based Social Systems*, vol. 7, pp. 3–18. Springer, Berlin/Heidelberg (2010)
55. Neviarouskaya, A., Tsetserukou, D., Prendinger, H., Kawakami, N., Tachi, S., Ishizuka, M.: Emerging system for affectively charged interpersonal communication. In: Proceedings of ICROS-SICE International Joint Conference, pp. 3376 –3381 (2009)
56. Nwana, H.S.: Software agents: an overview. The Knowledge Engineering Review **11**(03), 205–244 (1996)
57. Paolucci, M., Marsero, M., Conte, R.: What is the use of gossip? a sensitivity analysis of the spreading of respectful reputation. In: Tools and Techniques for Social Science Simulation, pp. 302–314. Physica-Verlag (2000)
58. Picard, R.W.: Toward agents that recognize emotion. In: Actes Proceedings IMAGINA, pp. 153–165 (1998)
59. de Pinninck, A.P., Sierra, C., Schorlemmer, W.M.: Distributed norm enforcement: Ostracism in open multi-agent systems. In: P. Casanovas, G. Sartor, N. Casellas, R. Rubino (eds.) Computable Models of the Law: Languages, Dialogues, Games, Ontologies, *Lecture Notes in Computer Science*, vol. 4884, pp. 275–290. Springer (2008)
60. Pujol, J.M.: Structure in artificial societies. Ph.D. thesis, Departament de Llenguatges i Sistemes Informàtics, Universitat Politènica de Catalunya (2006)
61. Sadri, F., Stathis, K., Toni, F.: Normative kgp agents. Computational & Mathematical Organization Theory **12**(2-3), 101–126 (2006)
62. Salazar, N., Rodríguez-Aguilar, J.A., Arcos, J.L.: Infection-based norm emergence in multi-agent complex networks. In: Proceeding of the 2008 conference on ECAI 2008, pp. 891–892. IOS Press, Amsterdam, The Netherlands, The Netherlands (2008)
63. Salazar, N., Rodríguez-Aguilar, J.A., Arcos, J.L.: Handling uncertainty in the emergence of social conventions. In: Third IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO'09), pp. 282–283 (2009)
64. Savarimuthu, B., Arulanandam, R., Purvis, M.: Aspects of active norm learning and the effect of lying on norm emergence in agent societies. In: D. Kinny, J. Hsu, G. Governatori, A. Ghose (eds.) Agents in Principle, Agents in Practice, *Lecture Notes in Computer Science*, vol. 7047, pp. 36–50. Springer Berlin / Heidelberg (2011)
65. Savarimuthu, B.T.R., Cranefield, S.: Norm creation, spreading and emergence: A survey of simulation models of norms in multi-agent systems. Multiagent and Grid Systems **7**(1), 21–54 (2011)
66. Searle, J.R.: Speech Acts. Cambridge University Press, Cambridge, UK. (1969)
67. Sen, S., Airiau, S.: Emergence of norms through social learning. In: Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI), pp. 1507–1512. AAAI Press (2007)
68. Shoham, Y., Tennenholtz, M.: Emergent conventions in multi-agent systems: Initial experimental results and observations. In: Proceedings of the Third International Conference on the Principles of Knowledge Representation and Reasoning (KR), pp. 225–231. Morgan Kaufmann, San Mateo, CA, USA (1992)
69. Shoham, Y., Tennenholtz, M.: On social laws for artificial agent societies: Off-line design. Artificial Intelligence **73**(1-2), 231–252 (1995)
70. Sills, D.L.: International Encyclopedia of the Social Sciences. New York (1968)
71. Smith, J.M., Harper, D.: Animal signals. Oxford University Press, USA (2003)
72. Spence, M.: Job market signaling. The quarterly journal of Economics **87**(3), 355–374 (1973)
73. Tuomela, R.: The Importance of Us: A Philosophical Study of Basic Social Notions. Stanford Series in Philosophy, Stanford University Press (1995)
74. Ullmann-Margalit, E.: The Emergence of Norms. Clarendon Press (1977)

75. Vázquez-Salceda, J.: Thesis: The role of norms and electronic institutions in multi-agent systems applied to complex domains. the harmonia framework. AI Communications **16**(3), 209–212 (2003)
76. Vázquez-Salceda, J., Aldewereld, H., Dignum, F.: Norms in multiagent systems: from theory to practice. Computer Systems: Science & Engineering **20**(4) (2005)
77. Verhagen, H.: Simulation of the Learning of Norms. Social Science Computer Review **19**(3), 296–306 (2001)
78. Villatoro, D., Sabater-Mir, J.: Categorizing social norms in a simulated resource gathering society. In: Coordination, Organizations, Institutions and Norms in Agent Systems IV: COIN 2008 International Workshops, Revised Selected Papers, pp. 235–249. Springer-Verlag, Berlin/Heidelberg (2009)
79. Villatoro, D., Sen, S., Sabater-Mir, J.: Topology and memory effect on convention emergence. In: WI-IAT '09: Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, pp. 233–240. IEEE Computer Society, Washington, DC, USA (2009)
80. Walker, A., Wooldridge, M.: Understanding the emergence of conventions in multi-agent systems. In: V. Lesser (ed.) Proceedings of the First International Conference on Multi-Agent Systems (ICMAS), pp. 384–389. AAAI Press, Menlo Park, California, USA (1995)
81. Watkins, C.J.C.H., Dayan, P.: Q-learning. Machine Learning **8**(3-4), 279–292 (1992)
82. Weisstein, E.W.: Torus. http://mathworld.wolfram.com/Torus.html (2010). Accessed on 01 December 2009
83. von Wright, G.H.: Deontic logic. Mind **60**(237), 1–15 (1951)
84. Yu, B., Singh, M.P.: Emergence of agent-based referral networks. In: Proceedings of the first international joint conference on Autonomous agents and multiagent systems (AAMAS), pp. 1268–1269. ACM, New York, NY, USA (2002)