Data-Driven Revision of Conditional Norms in Multi-Agent Systems (Extended Abstract)*

Davide Dell’Anna1, Natasha Alechina2, Fabiano Dalpiaz2 and Mehdi Dastani2 and Brian Logan2,3

1Delft University of Technology, Delft, The Netherlands
2Utrecht University, Utrecht, The Netherlands
3University of Aberdeen, Aberdeen, United Kingdom
d.dellanna@tudelft.nl, {n.alechina, f.dalpiaz, m.dastani, b.logan}@uu.nl

Abstract

In multi-agent systems, norm enforcement is a mechanism for steering the behavior of individual agents in order to achieve desired system-level objectives. Due to the dynamics of multi-agent systems, however, it is hard to design norms that guarantee the achievement of the objectives in every operating context. Also, these objectives may change over time, thereby making previously defined norms ineffective. In this paper, we investigate the use of system execution data to automatically synthesise and revise conditional prohibitions with deadlines, a type of norms aimed at preventing agents from exhibiting certain patterns of behaviors. We propose DDNR (Data-Driven Norm Revision), a data-driven approach to norm revision that synthesises revised norms with respect to a data set of traces describing the behavior of the agents in the system. We evaluate DDNR using a state-of-the-art, off-the-shelf urban traffic simulator. The results show that DDNR synthesises revised norms that are significantly more accurate than the original norms in distinguishing adequate and inadequate behaviors for the achievement of the system-level objectives.

1 Introduction

Multi-agent systems (MASs) comprise autonomous agents that interact in a shared environment [Wooldridge, 2009]. For example, a smart traffic system is a MAS that includes autonomous agents like cars, pedestrians, smart traffic lights, etc. To achieve the system-level objectives of a MAS, the behavior of the autonomous agents should be controlled and coordinated [Bulling and Dastani, 2016]. Norms and their enforcement are one way to do this without limiting the autonomy of the agents in a MAS [Chopra et al., 2018; Tinnemeier et al., 2009; Testerink et al., 2016]. Similar to our society, norms can be viewed as standards of behavior specifying that certain states or sequences of actions in a MAS should occur (obligations) or should not occur (prohibitions) in order for the objectives of the MAS to be achieved [Boella and van der Torre, 2004].

For many applications, it is assumed that the behavior of the agents and the norms enforced in the MAS will guarantee the achievement of the system’s objectives [Dastani et al., 2009]. This is possible, for example, when the MAS designer and agent designers are the same people, or when the agent’s behavior is enforced via the regimentation of norms, so that the agents cannot deviate from the intended behavior.

Regimentation, however, is particularly difficult in open MASs [Dastani et al., 2004; Artikis and Pitt, 2001], where agents can freely enter or leave the system, and their internal architecture is not known to the MAS designer. Furthermore, without awareness of how the agents internally behave, the MAS designer cannot fully predict how the agents will react to the norms, making it impossible, or computationally infeasible, to design norms that guarantee the satisfaction of the system’s objectives in every possible combination of the agents’ behaviors. Also, the MAS objectives may change over time, and the designed norms may become outdated and ineffective for the new objectives [Bicchieri, 2005].

To cope with these issues, norms need to be continuously evaluated, and possibly revised when they become inadequate for achieving the MAS objectives [Knobbbout et al., 2014; Knobbout et al., 2016; Dell’Anna et al., 2020].

In this paper, we make the following contributions. We introduce DDNR, a novel Data-Driven Norm Revision approach. DDNR revises norms from data describing the behavior of the agents in a MAS at run-time, and does not assume control over agents’ design nor requires regimentation. We evaluate the complexity of DDNR (see full paper [Dell’Anna et al., 2022a]). We apply and experimentally evaluate a Java implementation of DDNR, available online [Dell’Anna et al., 2022b], on an agent-based traffic simulation where norms regulate the behavior (maximum speed and minimum safety distance) of vehicles on a highway.

2 Norm Revision in Normative MASs

As an illustrative example, we use the highway section shown in Figure 1.

We focus on conditional prohibitions with deadlines, which express behavioral properties [Tinnemeier et al., 2009]. A conditional prohibition (over a finite propositional
Monitoring and Norm Enforcement
Data-Driven Norm Revision
Labeled traces
Norms
Updates
Collects traces for
Monitoring and Norm Enforcement

Initial design
Input to
Norms

Communicates norms and sanctions to the agents

agent entering the system

monitors (e.g., via runtime monitoring [Alechina
et al., 2014]) agent behavior, and stores the collected data in
a database in the form of a data set of finite traces. A trace
in the running example represents the car journey through the
highway, and it is generated by the actions of a vehicle. An
example of a trace γ composed of 10 states is the following,
where km
i
indicates that the vehicle reached the i
th
km of the
highway, sp
x
indicates that the vehicle’s speed is higher than a
certain speed x in km/h, and car indicates the vehicle’s type.

\[
\gamma = \{(km_1, sp_{30}, car), (km_2, sp_{22}, car), \ldots, (km_9, sp_{18}, car), (km_{10}, sp_{14}, car)\}
\]

(a)

(b)

Figure 1: A Normative Multi-Agent System, where norms are used to
to control the behavior of the autonomous agents (small black rect-
angles resembling vehicles) in the MAS. The Data-Driven Norm Revision
module revises the norms based on the collected data la-
bled by the MAS objectives evaluator (labeled traces). The MAS
objectives evaluator provides a labeling of the monitored agents’ be-
behaviors (collected traces) w.r.t. the MAS objectives.

tures evaluator component, which labels each trace as either positive or negative with respect to the MAS objectives. The
MAS objectives evaluator is assumed to be an external component, either human or automated, beyond the scope of this paper.

This paper focuses on the Data-Driven Norm Revision module (the green box in Figure 1), which revises the cur-
rently enforced norms so to ensure that (i) traces (behaviors) that are labeled as negative by the MAS objectives evaluator
are prohibited by the revised norms, and (ii) traces that are labeled as positive by the MAS objectives evaluator are not
prohibited by the revised norms. The Data-Driven Norm Revision module is agnostic to the MAS objectives, for it only
relies on labeled traces provided by the MAS objectives evaluator, which are considered as ground truth. This guaran-
tees that the proposed Data-Driven Norm Revision module is data driven and supports those cases where the MAS objec-
tives do not correspond directly to properties expressible in the
language L is a tuple (ϕC, ϕP, ϕD), where ϕC, ϕP and ϕD
are propositional formulas, expressed in Disjunctive Normal Form (DNF), over L. In this expression, ϕC is the (detach-
ment) condition of the norm, ϕD is the deadline, and ϕP is a target state that is prohibited to occur after a state where
the condition of the norm ϕC holds, and before a state where the deadline ϕD holds (the norm “expires”).

Example. The norm “if a car enters the 2nd km of the highway, it is prohibited from driving faster than 70 km/h until it
reaches the 7th km of the highway” can be represented as a conditional prohibition (km
2
∧ car, sp
70
, km
7
).

In our framework, a Monitoring and Norm Enforcement component monitors (e.g., via runtime monitoring [Alechina
et al., 2014]) agent behavior, and stores the collected data in a database in the form of a data set of finite traces. A trace
in the running example represents the car journey through the highway, and it is generated by the actions of a vehicle. An
example of a trace γ composed of 10 states is the following, where km
i
indicates that the vehicle reached the i
th
km of the
highway, sp
x
indicates that the vehicle’s speed is higher than a
certain speed x in km/h, and car indicates the vehicle’s type.

\[
\gamma = \{(km_1, sp_{30}, car), (km_2, sp_{22}, car), \ldots, (km_9, sp_{18}, car), (km_{10}, sp_{14}, car)\}
\]

Algorithm 1 DDNR

1: Input: data set Γ; set of norms N; list T of |N| types of
revisions, one for each norm in \(N\); list V of |N| triples
(\(V^C_n\), \(V^P_n\), \(V^D_n\)) of propositional variables, one for each
norm n in \(N\); number of samples k
2: Output: ordered list of revised norms selected
3: \(R(\mathcal{N}) \leftarrow [\] \quad \triangleright Synthesis step
4: \(\mathcal{R}(\mathcal{N}) \leftarrow [\) \quad \triangleright empty list
5: for all norm n \in N do
6: \(\mathcal{R}(\mathcal{N}).\text{APPEND}(\text{SYNTHESIS}(\Gamma, n, T_n, V^C_n, V^P_n, V^D_n))\) \quad \triangleright Selection step
7: candidates \leftarrow \text{RANDOMSAMPLE}(\mathcal{R}(\mathcal{N}), k)
8: return \text{argmax}_{\text{cand} \in \text{candidates} \cup \{N\}} \text{ML-ACC(cand, } \Gamma)\)

3 DDNR: Data-Driven Norm Revision

Given a set of norms \(N\) and a data set \(T\) of traces labeled w.r.t. the MAS objectives, \(DDNR\) (summarized in Algorithm 1) syntheses revised norms that are better aligned with the
MAS objectives with respect to \(G\).

\(DDNR\) consists of two steps: the synthesis step and the selection step. These are briefly described below for the case of revision of one norm. An in-depth technical description
of \(DDNR\), also for multiple norms, can be found in the full
version of the paper [Dell’Anna et al., 2022a].

3.1 Revising the Norm: the Synthesis Step

Consider a norm \(n = (\phi_C, \phi_P, \phi_D)\) to be revised. We distinguish three types of revisions of a norm: alteration, weakening,
and strengthening.

Alteration – Prohibiting Different Behaviors

An alteration of a norm \(n\) is a new norm \(n’\) that prohibits a
different set of behaviors than \(n\). An alteration of \(n\) can be realized by making at least one of the components of \(n\),
that is the condition, the target state or the deadline, either
more or less specific. For example, an alteration of \(n\) is a new
norm \(n’ = (\phi’C, \phi’P, \phi’D)\) such that \(\phi’C\) is less specific than
\(\phi_C\), while \(\phi’P\) and \(\phi’D\) are more specific than \(\phi_P\) and \(\phi_D\), respectively.
Technical details and algorithms for the synthesis of these three types of revisions can be found in [Dell’Anna et al., 2022a]. Figure 2 provides examples of the types of norms that DDNR allows to synthesise from a data set of traces $\Gamma$.  

<table>
<thead>
<tr>
<th>Data set $\Gamma$</th>
<th>Norm $n$</th>
<th>Examples of weakening of $n$:</th>
<th>Examples of strengthening of $n$:</th>
<th>Examples of alteration of $n$:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trace</td>
<td>$s_1$</td>
<td>$s_2$</td>
<td>$s_3$</td>
<td>$s_4$</td>
</tr>
<tr>
<td>Trace</td>
<td>$s_1$</td>
<td>$s_2$</td>
<td>$s_3$</td>
<td>$s_4$</td>
</tr>
<tr>
<td>Trace</td>
<td>$s_1$</td>
<td>$s_2$</td>
<td>$s_3$</td>
<td>$s_4$</td>
</tr>
<tr>
<td>Trace</td>
<td>$s_1$</td>
<td>$s_2$</td>
<td>$s_3$</td>
<td>$s_4$</td>
</tr>
<tr>
<td>Trace</td>
<td>$s_1$</td>
<td>$s_2$</td>
<td>$s_3$</td>
<td>$s_4$</td>
</tr>
<tr>
<td>Trace</td>
<td>$s_1$</td>
<td>$s_2$</td>
<td>$s_3$</td>
<td>$s_4$</td>
</tr>
<tr>
<td>Trace</td>
<td>$s_1$</td>
<td>$s_2$</td>
<td>$s_3$</td>
<td>$s_4$</td>
</tr>
</tbody>
</table>

Figure 2: Data set $\Gamma$ composed of 8 finite traces, each made of 3 states (columns $s_1$-$s_3$). States are colored based on their types (more details in [Dell’Anna et al., 2022a]). On the right, traces are labeled according to a norm $n$ and to 7 examples of revisions of $n$ obtained from the synthesis step: the cells containing ✓ indicate that the corresponding trace is norm-violating, empty cells indicate that the trace is norm-compliant.

3.2 Choosing the New Norm: the Selection Step

The selection step chooses a new norm from the set $R(n)$ of candidate revisions of $n$ synthesised during the synthesis step.

The relationship between the classification of traces according to a norm (i.e., whether the trace is classified as norm-compliant or norm-violating) and the correct classification of the traces according to the MAS objectives labeling can be described via a confusion matrix. Each cell $(i, j)$ in the matrix contains the number of traces in the data set $\Gamma$ that are classified as $i$ by the MAS objectives evaluator and as $j$ by the norm.

By analysing a confusion matrix that characterizes the traces in a data set w.r.t. the classification provided by a norm, we can determine the number of classification errors of a norm, the type of errors (i.e., whether negative traces are considered compliant more often than positive traces are considered violating), and we can determine whether a norm is better (aligned with the MAS objectives) than another.

We characterize the concept of alignment of a norm with the MAS objectives using the accuracy metric, defined as $\text{acc}(n, \Gamma) = \frac{TP + TN}{|\Gamma|}$, with $TP$ and $TN$ being the number of true positives and true negatives obtained with a norm $n$ on the data set of traces $\Gamma$.

Therefore, the selection step aims at choosing the revision of $n$ with highest accuracy, i.e., given $R(n)$, we choose, as a revision of $n$, the norm $n^* = \arg\max_{n' \in R(n)} \text{acc}(n', \Gamma)$.

4 Empirical Evaluation

We make use of a traffic simulation of the highway scenario described in Section 2 to generate a data set of traces describing the behavior of agents in a MAS. We implement the scenario with the SUMO traffic simulator [Krajzewicz et al., 2012], considering two types of vehicles (cars and trucks, each with their own properties such as maximum speed, acceleration, etc.). We use a population of agents randomly and uniformly distributed among cars and trucks. 75% of the agents are always compliant with the enforced norms, while the remaining 25% will ignore them (can afford the sanction resulting from violation) and focus on maximizing speed.

We execute 100 independent traffic simulations. In each simulation $i$, we enforce a norm $n_i$ randomly generated. We run every simulation $i$ until 1,500 vehicles drive through the highway section under the enforcement of $n_i$. Since the behavior of each vehicle corresponds to an execution trace (e.g., trace in Eq. 1), each simulation generates 1,500 traces.
norms were already too weak and did not need further weakening the number of negative traces was low (i.e., the original norms. The improvement with weakening led to norms with higher accuracy than the original norms, and shows the accuracy change in blue, and the resulting accuracy after revision in red. Each trace is labeled by the MAS objectives evaluator as positive if the maximum emission of the vehicle from the beginning to the end of the highway section is below a threshold $t_{\text{co}_2} = 100$ g/s and the travel time is below a threshold $t_{\text{str}} = 450$ s (the time it takes to drive for 10 km at 80 km/h), and negative otherwise. Given the resulting labeled data set $\Gamma$, we execute DDNR for the three types of revisions. For each of the 100 norms initially enforced, we obtain 100 weakened norms, 100 strengthened norms and 100 altered norms. We analyze the accuracy of these revised norms in comparison with the 100 original norms.

Figure 3 compares the accuracy of the original norms with that of the revised norms, and shows the accuracy change when performing weakening, strengthening or alteration. All operations led to norms with significantly higher accuracy than the original norms. The improvement with weakening has a smaller magnitude, with a negligible effect size $d_{\text{Cohen}} = 0.087$, because in the data set obtained from simulation the number of negative traces was low (i.e., the original norms were already too weak and did not need further weakening). In the case of strengthening, instead, the average improvement of accuracy is around 20%, with a large effect size $d_{\text{Cohen}} = 1.115$. Finally, with alteration, the average accuracy improves even more, with a large improvement of around 24% ($d_{\text{Cohen}} = 1.275$). Note that half of the new norms have an accuracy higher than 76%, and 25% of the norms have an accuracy higher than 89%.

In the full version of this paper [Dell’Anna et al., 2022a], we report on an in-depth analysis of these and additional experimental results. Experiments reported in the full paper also include results for the revision of multiple norms and a study on how well the revised norms generalize to previously unseen traces. Overall, results show that the revised norms are significantly more accurate (aligned with the MAS objectives) than the original norms, exhibiting an average improvement of accuracy on the given data set of traces of about 30% and an average improvement of accuracy of about 13% on unseen traces.

5 Conclusions and Future Work

We investigated the problem of norm revision in contexts where the internals of the agents in a MAS are unknown and where norms are expressed in a different language from that of the MAS objectives that they intend to bring about. We presented results regarding the automated synthesis and revision of conditional norms (prohibitions) with deadlines w.r.t. a set of observed traces representing the behavior of the agents in the MAS. The traces are partitioned into positive and negative ones, depending on whether each helps or hurts MAS objectives. Besides a boolean evaluation, the revision mechanism possesses no information about the relationship between a trace and the objectives. We proposed DDNR (Data-Driven Norm Revision): a practical heuristic approach to obtain approximate revisions of the conditional norms. We applied DDNR to a traffic simulation. Results show that the revised norms are significantly more accurate (aligned with the MAS objectives) than the original norms. In future work, we intend to embed DDNR in a runtime supervision framework presented in earlier work [Dell’Anna et al., 2019; Dell’Anna et al., 2020] that continuously monitors the system’s execution and, based on probabilistic strategies, suggests how to revise the norms (i.e., whether to alter, weaken or strengthen them) to continuously guarantee the achievement of the MAS objectives. Considering the relative importance of different states and propositions in revising norms is another future direction of our work. Similarly, we also intend to extend the proposal to support multiple system objectives, and to more fine-grained evaluations (as opposed to the considered boolean evaluation) of the traces with respect to the objectives.

References


