

Interpreting Models of Social Group Interactions in Meetings with Probabilistic Model Checking

Oana Andrei
University of Glasgow
Glasgow, UK
oana.andrei@glasgow.ac.uk

Gabriel Murray
University of the Fraser Valley
Abbotsford, B.C., Canada
gabriel.murray@ufv.ca

ABSTRACT

A major challenge in Computational Social Science consists in modelling and explaining the temporal dynamics of human communication. Understanding small group interactions can help shed light on sociological and social psychological questions relating to human communications. Previous work showed how Markov rewards models can be used to analyse group interaction in meeting. We explore further the potential of these models by formulating queries over interaction as probabilistic temporal logic properties and analysing them with probabilistic model checking. For this study, we analyse a dataset taken from a standard corpus of scenario and non-scenario meetings and demonstrate the expressiveness of our approach to validate expected interactions and identify patterns of interest.

KEYWORDS

Small groups, social sequences, Markov rewards models, probabilistic temporal logic, probabilistic model checking

ACM Reference Format:

Oana Andrei and Gabriel Murray. 2018. Interpreting Models of Social Group Interactions in Meetings with Probabilistic Model Checking. In *Proceedings of Group Interaction Frontiers in Technology (GIFT'18)*. ACM, New York, NY, USA, 7 pages.

1 INTRODUCTION

A major challenge in Computational Social Science [6, 12, 15] consists in modelling and explaining the temporal dynamics of human communications. Which interactions lead to more successful communication or productive meetings? How can we infer temporal models of interactions? How can we explain what these temporal interaction really mean? Current statistical analysis techniques do not explore the full temporal aspect of time-series data generated by interactive systems, and certainly they do not address complex queries involving temporal dependencies.

We investigate Markov rewards models (also called discrete-time Markov chains with rewards) for human-human interactions in social group meetings and how to interpret them. We identify various queries predicating over the temporal interactions between different roles, the impact of different sentiments in interactions or in decision making, causality between particular states, etc. We use probabilistic computational tree logic (PCTL) with rewards [4, 11], which is a type of probabilistic temporal logic variant, to formalize these queries. We then use the PRISM tool [11], a symbolic probabilistic model checker, to analyse the formal queries and thus interpret the temporal interaction models. Probabilistic model checking [4] is a well-established verification technique that explores all possible states of a Markov model in a systematic and exhaustive manner, and computes the probability that a temporal property of

the system under analysis holds. We can ask queries such as ‘*What is the average count of the project manager’s interventions until a decision is taken?*’, ‘*What is the probability of a decision to be taken without anybody commenting about their understanding?*’, or ‘*What is the average interaction count from one decision to another decision without a negative sentiment being expressed in the interim?*’. Figure 1 illustrates the method we propose for probabilistic modeling analysis of social group behaviour.

The main contribution of this paper consists in empirically demonstrating the expressiveness of probabilistic temporal logic properties and probabilistic model checking for the analysis of temporal dynamics of social group interactions in meetings.

2 RELATED WORK

Our work is most closely related to the Markov Rewards Model by Murray [13, 14] for analyzing and querying social sequences. In that work, social interactions are represented as a sequence of states, and particular states are associated with rewards or costs that are dependant on the query being asked. A Value Iteration algorithm is then used to estimate the expected value of every state, with a state’s value indicating how it is related to the outcome of interest being queried. In our work, we will use the same state representation as Murray, but show that our probabilistic model checking framework allows us to ask queries that would be difficult or impossible to ask in the Markov Rewards Model framework. More generally, our approach is an example of *social sequence analysis* [7], where the goal is to analyze patterns in social sequences or to compare social sequences to one another. These social sequences might unfold at the macro scale (over days or weeks) or at the micro scale (over minutes or hours), and the present work is concerned with social sequences at the micro scale.

The past decade has seen an increasing amount of work on developing technologies for supporting meetings, including the use of machine learning for making predictions on meeting data. This includes detection of decision items [10] and classification of dialogue act types [8], in addition to predictions for many other meeting phenomena [16]. The field of Social Signal Processing (SSP) consists of work that examines social interaction through primarily *nonverbal* cues [18], such as gesture, gaze, and prosody. There is also a growing inter-disciplinary field of *meeting science* that aims to understand the processes that take place before, during, and after meetings [1].

3 CORPUS

The dataset used in this paper is the Augmented Multimodal Interaction (AMI) meeting corpus [5]. Each meeting group in the corpus consists of four people, and the group completes a sequence of

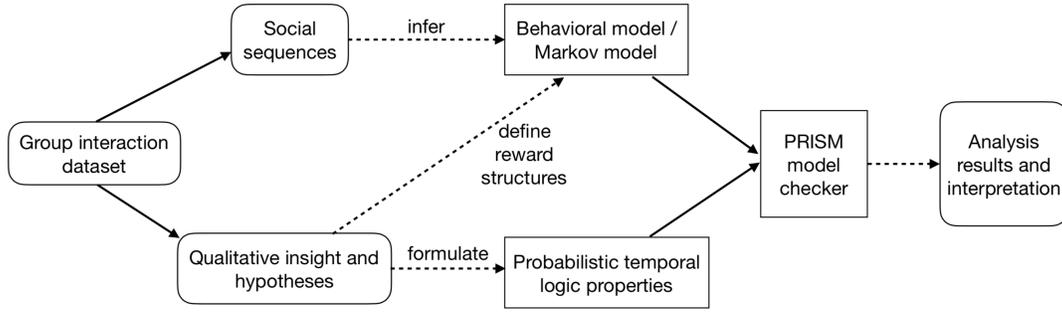


Figure 1: Overall process of modeling and analysis of group interactions

four meetings where they are role-playing as members of a company that is designing and marketing a product. Each person in the group is assigned a role; the roles are Project Manager (PM), Marketing Expert (ME), User Interface Designer (UI), and Industrial Designer (ID). Despite the artificial scenario and the assigned roles, the speech is spontaneous and unscripted, and each group is free to make decisions as they see fit. We discuss further aspects of the corpus in Section 4.1, where we describe the state representation used in this work.

4 PROBABILISTIC TEMPORAL MODELLING AND ANALYSIS OF INTERACTION

In this section we describe the state representation used in our Markov models, the probabilistic temporal logic properties and reward structures used for formalising queries about group interactions captured by the Markov model, and the probabilistic model checker PRISM used for formally analysing these queries.

4.1 Markov models of social group interactions

In our representation of social sequences in meeting, each state is labelled by a 5-tuple consisting of the following information:

- (1) the participant’s role in the group: PM (Project Manager), ME (Marketing Expert), UI (User Interface Designer), and ID (Industrial Designer);
- (2) the dialogue act type taking one of the 15 values listed and briefly described in Table 1;
- (3) the sentiment being expressed: nosentiment, positive, negative, posneg (both);
- (4) whether the utterance involves a decision: nodecision, decision;
- (5) whether the utterance involves an action item: noaction, yesaction.

In addition to the complex states described in the preceding section, there are START and STOP labeled states representing the beginning and the end of a meeting.

Example states include the following:

- <PM-bck-positive-nodecision-noaction> describes the situation where the project manager makes a positive backchannel comment, unrelated to a decision or action;
- <PM-el.ass-nosentiment-nodecision-yesaction> represents the project manager eliciting feedback about an action item;

Table 1: Dialogue Act Types

ID	Description
fra	fragment
bck	backchannel
stl	stall
inf	inform
el.inf	elicit inform
sug	suggest
off	offer
el.sug	elicit offer or suggestion
ass	assessment
und	comment about understanding
el.ass	elicit assessment
el.und	elicit comment about understanding
be.pos	be positive
be.neg	be negative
oth	other

- <UI-sug-nosentiment-decision-noaction> represents the UI expert making a suggestion about a decision item.

The Markov aspect of the Markov models is that the probability of a given state depends only on the preceding state in the sequence. The state transition probabilities are estimated directly from the transition counts in the data. This way we obtain a discrete-time Markov model of the behaviour seen in the meeting data, where the states labels and the transition probability function are defined as above, the initial state is labelled by START. A *path* in a Markov model is a non-empty sequence of states such that the transition probability from one state to the next one in the sequence is strictly greater than zero.

4.2 Probabilistic temporal logic and model checking

Probabilistic model checking is a technique for modelling and analysing stochastic systems, usually focused on investigating correctness properties of the real-life system. It requires an abstract, high-level description of the system and specifications of the properties expressed in a suitable temporal logic. In the first step a probabilistic model checker tool builds a model of the system from

its description, typically a Markov model (e.g., discrete time Markov chain, continuous time Markov chain, or Markov decision process). In the second step, the tool uses model checking algorithms to verify automatically if a temporal logic property is satisfied or not, or to compute the probability of a temporal logic formula to hold. These model checking algorithms explore the model in an systematic and exhaustive way.

Probabilistic Computation Tree Logic (PCTL) [4, 11] is a probabilistic branching-time temporal logic that allows one to express a probability measure of the satisfaction of a temporal property by a state of a discrete-time Markov model. The syntax is the following:

State formulae:

$$\Phi ::= \text{true} \mid a \mid \neg \Phi \mid \Phi \wedge \Phi \mid P_{\triangleright \triangleleft p}[\Psi] \mid S_{\triangleright \triangleleft p}[\Phi]$$

Path formulae:

$$\Psi ::= X\Phi \mid \Phi \cup^{\leq N} \Phi$$

where a represents an atomic proposition, $\triangleright \triangleleft \in \{\leq, <, \geq, >\}$, $p \in [0, 1]$, and $N \in \mathbb{N} \cup \{\infty\}$. The operators X and U are called the **next** and the **Until** operators respectively.

PCTL formulae (or properties) are interpreted over states of a Markov model, with state formulae Φ evaluated over states and path formulae Ψ over paths. We say that a Markov model satisfies a state formulae Φ if the initial state of model satisfies Φ . We denote by $s \models \Phi$ that state s satisfies Φ (or Φ is evaluated to true in state s) and this satisfaction relation is defined inductively as follows:

- $s \models \text{true}$ is always true;
- $s \models a$ if and only if a is an atomic proposition labelling the state s ;
- $s \models \neg \Phi$ if and only if $s \models \Phi$ is false;
- $s \models \Phi_1 \wedge \Phi_2$ if and only if $s \models \Phi_1$ and $s \models \Phi_2$;
- $s \models P_{\triangleright \triangleleft p}[\Psi]$ if and only if the probability that the path formula Ψ is satisfied by the paths starting from state s meets the bound $\triangleright \triangleleft p$;
- $s \models S_{\triangleright \triangleleft p}[\Phi]$ if and only if the steady-state (or long-run) probability of being in a state that satisfies the state formula Φ meets the bound $\triangleright \triangleleft p$.

For a path π starting from a state s , we define the satisfaction relation $\pi \models \Psi$ as follows:

- $\pi \models X\Phi$ is true if and only if Φ is satisfied in the *next* state following s in the path π ;
- $\pi \models \Phi_1 \cup^{\leq N} \Phi_2$ is true if and only if Φ_2 is satisfied within N time steps and Φ_1 is true up *until* that point where Φ_2 is satisfied for the first time.

The syntax above includes only a minimal set of operators; the propositional operators *false*, disjunction \vee and implication \implies can be derived. Two common derived path operators are: the *eventually* operator F where $F^{\leq n} \Phi \equiv \text{true} \cup^{\leq n} \Phi$ and the *always* operator G where $G\Psi \equiv \neg(F\neg\Psi)$. If $N = \infty$, i.e., the until operator U is not bounded, then the superscript is omitted.

For example, how do we check whether the probability of reaching a *yesaction* within 50 utterances while the sentiment being expressed is not a *positive* one is greater than 0.75? The corresponding PCTL property represented as $P_{\geq 0.75}[\neg \text{“positive”} \cup^{\leq 50} \text{“yesaction”}]$. The model checking algorithm computes the reachability probability for all states satisfying the atomic proposition *“yesaction”* provided that all previous states visited do not satisfy the atomic

proposition *“positive”*; if the resulting probability is greater than 0.75 then the model checking problem returns true; otherwise it returns false.

PRISM is a probabilistic model checker [11] used for formal modelling and analysis of systems that exhibit random or probabilistic behaviour. Its high-level state-based modelling language supports a variety of probabilistic models, including discrete-time Markov chains. In PRISM we can replace the bounds $\triangleright \triangleleft p$ in the properties with $=?$ and thus obtain the numerical value that makes the property true. PRISM also allows models to be augmented with reward structures, which assign positive real values to states and/or transitions for the purpose of reasoning over expected or average values of these rewards. In PRISM we can specify the following reward-based temporal properties:

- $R_{reward=?} [C^{\leq N}]$ in a state s computes the expected value of the reward named *reward* accumulated along all paths starting from s within N time-steps.
- $R_{reward=?} [F\Phi]$ in a state s computes the expected value of the reward named *reward* accumulated along all paths starting from s until the state formula Φ is satisfied.

In PRISM, filters check for properties that hold when starting from sets of states satisfying given propositions. In this paper we use the filter operators *state* and *avg* in the following two types of properties:

- $\text{filter}(\text{state}, \Phi, \text{cond1})$ evaluates the satisfaction of the state formula Φ in the state uniquely identified by the Boolean proposition *cond1*;
- $\text{filter}(\text{avg}, \Phi, \text{cond2})$ computes the average over all states where *cond2* is true.

In the following, for convenience, we refer to PCTL properties with or without rewards simply as properties or queries, though strictly they also include PRISM operators.

5 EXPERIMENTS AND RESULTS

In this section, we first define the behavioural model used, followed by a set of queries, their encoding as probabilistic temporal logic properties, and their results, which demonstrate the flexibility and expressiveness of the method presented in this paper.

5.1 Defining a behavioural model of social group interactions

The behavioural model is a Markov rewards model initially inferred as described in Section 4.1 to which we add labels and reward structure definitions as required by the queries. In our case the atomic propositions associated with each state are the state labels and the individual particles composing the state label.

The PRISM model encoding the Markov model for the input data set considered for this paper as well as the PRISM properties analysed later in this paper are available at http://www.dcs.gla.ac.uk/~oandrei/resources/imsgi_gift18. The PRISM model has a relatively small state space of 196 reachable states (out of 269 states in total) and 4002 transitions, therefore the model checking process for one temporal property is not time-consuming (under 0.1 seconds for all instances of the properties listed in the next section).

We defined the following reward structures:

- `r_Steps` assigns a value of 1 to each transition or time-step. We use this reward structure when computing the average number of time-steps (i.e., interactions) from one state to another state.
- `r_roleLabel` is a template reward structure which assigns a value of 1 to each state labelled by `roleLabel`. Since there are four participant roles in the data set, we have four reward structures `r_PM`, `r_ME`, `r_UI`, and `r_ID`. For example, we can use the reward structure `r_PM` to compute the average visit count to a PM state until reaching a state tuple which includes decision, i.e., the average number of PM interventions until a decision is made.
- `r_roleLabel_decision` is also a template reward structure which assigns a value of one to each state labelled by one instance of `roleLabel` (either PM, ME, UI, or ID) and by decision. For example, we can use the reward structure `r_PM_decision` to compute the average number of PM interventions concerning a decision until reaching a decision state where ME intervenes.

5.2 Querying the Markov model

We use the command line of the PRISM tool to execute each of the queries presented in this section through the probabilistic model checking engine and export the results. For some of the PRISM properties below we make the following notation for the sake of brevity. We use the placeholder `roleLabel` to be instantiated with any of the roles PM, ME, UI, or ID. The atomic proposition $y=j$ refers to the state variable y in the PRISM model with the identifier j ; in this case j takes values from 0 to 268.

5.2.1 Queries for validating the model. We first start with examples of queries and results that confirm our expectations about meetings generally and the AMI scenario specifically. For example, some of the results reflect the fact that project managers (PM) tend to begin meetings, and – in the AMI scenario, at least – are the most active participants. Some of the results of this first set of queries are merely artifacts of the AMI scenario, and in particular of the fact that participants are assigned clearly-defined roles and have to progress through distinct phases of a role-playing exercise.

We then move on to queries and results that generate more insight into meeting interactions.

Q1: How long does it typically take in a meeting before each type of role has participated?

These queries are encoded in PRISM as:

```
R{"r_Steps"}=?[F "PM"]
R{"r_Steps"}=?[F "ME"]
R{"r_Steps"}=?[F "ID"]
R{"r_Steps"}=?[F "UI"]
```

Each of the PRISM queries above computes the average accumulated number of time steps (or interactions) it takes to reach a state corresponding to a particular role. The actual average number of steps is computed using the transition reward `r_Steps`.

The analysis results are 2.13 time steps for PM, 5.26 for ME, 5.99 for ID, and 6.03 for UI. This is an intuitive (and expected) result, showing that the project manager (team leader) tends to begin the meeting discussions, but also that all members participate early on in the discussion.

Q2: How long does it typically take in a meeting before each type of non-PM role has participated after a Project Manager?

The PRISM properties encoding of Q2 are:

```
filter(avg, R{"r_Steps"}=?[F "ME"], "PM")
filter(avg, R{"r_Steps"}=?[F "ID"], "PM")
filter(avg, R{"r_Steps"}=?[F "UI"], "PM")
```

Such properties compute the average number of time steps taken from a PM intervention until a specific non-PM participant intervenes. The model checking results, in increasing order, are: 4.82 for ME, 5.28 for ID, 5.81 for UI. This shows that ME is quicker in reacting after a PM intervention than ID and UI. Since the participant roles are specific to the AMI corpus, these results are likely just reflecting the structure of the AMI scenario itself.

Q3: At the time that a first decision has been made, what is the proportion of activity by each participant? In other words, this query refers to the average number of times a type of participant role intervenes until eventually a decision is made. Let $\chi_3(\text{roleLabel})$ denote the PRISM property computing the average visit counts to a `roleLabel` state until a decision is made:

```
R{"r_roleLabel"}=?[F "decision"]
```

Then the PRISM property encoding Q3 is:

$$\chi_3(\text{roleLabel}) / (\chi_3(\text{PM}) + \chi_3(\text{ME}) + \chi_3(\text{UI}) + \chi_3(\text{ID}))$$

Checking this property instantiated with each of the four roles, we obtain that PM participates 32%, ME 24%, while UI and ID are participating in equal measure at 22%. This results reflect the fact that project managers tend to be more dominant in the meeting discussions, and in particular in regards to decision-making.

Q4: How many times in average a PM (or some other role) is involved in decision-making within 100 time steps?

Let $\chi_4(\text{roleLabel})$ denote the PRISM property that computes the average visit counts to states where `roleLabel` made a decision within 100 time steps:

```
R{"r_roleLabel_decision"}=?[C<=100]
```

Then the PRISM property encoding Q4 is:

$$\chi_4(\text{roleLabel}) / (\chi_4(\text{PM}) + \chi_4(\text{ME}) + \chi_4(\text{UI}) + \chi_4(\text{ID}))$$

After checking the four instances of this property, we obtain the following results: 86% for PM, 9% for UI, 3% for ID, and 1% for ME. As expected, project managers are making the majority of decisions, and the differences between the other three roles are likely an artifact of the AMI scenario.

Q5: Which type of non-PM roles is more participatory following a PM within 100 time steps?

The PRISM property encoding this query averages over all PM states the visit counts to `roleLabel` within 100 time steps:

```
filter(avg, R{"r_roleLabel"}=?[C<=100], "PM")
```

and the results of model checking it are: 36% for ME, 33% for ID, and 32% for UI.

This shows that the non-PM roles are approximately equally likely to participate after the PM, with the ME being slightly more frequent. Again, this may be an artifact of the AMI scenario.

Q6: Which roles with positive sentiment have the highest probability in the long-run?

The PRISM property encoding this query looks at the probability in the long-run to be in a particular type of role with a positive sentiment:

S=?["roleLabel" & "positive"]

and the results are as follows: 34% for PM, 32% for ME, 18% for ID, 16% for UI.

These results largely reflect the fact that the PM tends to be most active person in the AMI meeting discussions.

5.2.2 *Queries for further exploration of interactions.* Many of the preceding sets of queries and results conform to our expectations about meeting behaviour and the AMI scenario. We now turn to a set of queries and results that generate more valuable insight into meeting interactions.

Q7: Which non-decision states are most valuable in contributing to decisions being made within 100 time-steps?

The PRISM property encoding this query computes the probability of reaching a decision state within 100 time-steps when starting from a specific non-decision state:

filter(state, P=?[F<=100 "decision"], (y=j)&"nodecision")

The top ten most valuable non-decision states (i.e., most likely to lead to a decision within 100 time-steps) are the following:

State Label	Result
PM_stl_positive_nodecision_noaction	0.2870
PM_fra_positive_nodecision_noaction	0.2699
ID_stl_positive_nodecision_noaction	0.2649
ID_bck_positive_nodecision_noaction	0.2486
PM_stl_negative_nodecision_noaction	0.2450
ME_stl_negative_nodecision_noaction	0.2420
PM_sug_positive_nodecision_noaction	0.2219
PM_inf_negative_nodecision_noaction	0.2204
PM_stl_posneg_nodecision_noaction	0.2187
PM_off_positive_nodecision_noaction	0.2142

The most noticeable trend is that states containing sentiment – both positive and negative – are highly associated with decision-making. A second trend is that non-decision states belonging to the PM are highly associated with decisions being made. Both of these findings are intuitive; for example, participants tend to express a variety of opinions before mutually deciding on a solution or course of action.

Q8: Which PM states tend to lead to more participation by non-PM participants within 50 time-steps?

The corresponding PRISM property for the ME role sets a reward of 1 for each visit of a ME state and hence computes the average visit counts to ME states within 50 time-steps when starting from a specific PM state.

filter(state, R{"r_ME"}=?[C<=50], (y=j) & "PM")

The top ten results in terms of average count of ME interventions are:

State Label	Result
PM_elUnd_negative_nodecision_noaction	12.7667
PM_bck_positive_nodecision_noaction	12.3441
PM_off_positive_nodecision_noaction	12.3024
PM_oth_positive_nodecision_noaction	12.2464
PM_ass_posneg_nodecision_noaction	12.1870
PM_off_negative_nodecision_noaction	12.0611
PM_beNeg_nosentiment_nodecision_noaction	11.9997
PM_bck_nosentiment_nodecision_noaction	11.9907
PM_elInf_positive_nodecision_noaction	11.9819
PM_ass_positive_nodecision_noaction	11.9564

The PRISM property for the UI role is:

filter(state, R{"r_UI"}=?[C<=50], (y=j) & "PM")

and the results in terms of average count of UI interventions are:

State Label	Result
PM_oth_negative_nodecision_noaction	10.7655
PM_ass_nosentiment_decision_noaction	10.7637
PM_und_positive_nodecision_noaction	10.7478
PM_elAss_negative_nodecision_noaction	10.7231
PM_bePos_positive_nodecision_noaction	10.7143
PM_elAss_nosentiment_decision_noaction	10.7064
PM_bck_nosentiment_nodecision_noaction	10.6959
PM_elInf_negative_nodecision_noaction	10.6893
PM_elInf_nosentiment_nodecision_noaction	10.6765
PM_elAss_nosentiment_nodecision_noaction	10.6639

The PRISM property for the ID role is:

filter(state, R{"r_ID"}=?[C<=50], (y=j) & "PM")

and the results in terms of average count of interventions are:

State Label	Result
PM_inf_nosentiment_decision_yesaction	11.3406
PM_elInf_nosentiment_decision_noaction	11.2798
PM_elAss_positive_decision_noaction	11.2502
PM_bck_positive_nodecision_noaction	11.0955
PM_sug_posneg_nodecision_noaction	11.0941
PM_elAss_nosentiment_decision_noaction	11.0753
PM_off_negative_nodecision_noaction	11.0621
PM_sug_positive_decision_noaction	11.0404
PM_elAss_positive_nodecision_noaction	11.0204
PM_elInf_nosentiment_nodecision_noaction	11.0080

These results tell us that the PM is particularly likely to get participation from other members when he or she explicitly seeks input (e.g. elAss and elInf dialogue act types) and when expressing sentiment.

Q9: Which non-sentiment states are highly associated with positive sentiment?

The PRISM property encoding this query looks at each state tuple with no sentiment being expressed and then computes the probability of the next state to include a positive sentiment:

filter(state,P=?[X "positive"], (y=j) & "nosentiment")

The top ten non-sentiment states most likely to be associated with positive sentiment in the next state are the following:

State Label	Result
PM_elAss_nosentiment_decision_noaction	0.1456
PM_elInf_nosentiment_decision_noaction	0.0825
ID_und_nosentiment_noddecision_noaction	0.0510
ME_beNeg_nosentiment_noddecision_noaction	0.0505
PM_off_nosentiment_noddecision_yesaction	0.0408
PM_elAss_nosentiment_noddecision_noaction	0.0319
ME_elAss_nosentiment_noddecision_noaction	0.0319
ME_elSug_nosentiment_noddecision_noaction	0.0312
ID_elUnd_nosentiment_noddecision_noaction	0.0306
ID_elSug_nosentiment_noddecision_noaction	0.0297

These results show that states containing dialogue acts that are explicitly eliciting information (e.g. elAss, elSug, elUnd, elInf) are likely to be followed by expressions of positive sentiment. In particular, the top state represents the PM explicitly seeking an assessment from one or more of the other group members, and this is very likely to be followed by a positive sentiment state.

Q10: Which non-sentiment states are highly associated with negative sentiment?

Similar to Q9, the PRISM property encoding Q10 is:

```
filter(state,P=?[X "negative"],(y=j) & "nosentiment")
```

The top ten non-sentiment states most highly associated with negative sentiment in the next state are the following:

State Label	Result
ME_elUnd_nosentiment_noddecision_noaction	0.01923
ID_elAss_nosentiment_noddecision_noaction	0.01064
UI_elSug_nosentiment_noddecision_noaction	0.01064
ME_bck_nosentiment_noddecision_noaction	0.01053
ME_inf_nosentiment_decision_noaction	0.01042
ME_off_nosentiment_noddecision_noaction	0.01010
PM_elSug_nosentiment_noddecision_noaction	0.00990
UI_inf_nosentiment_decision_noaction	0.00962

Interestingly, states that explicitly elicit information and belong to somebody other than the PM are associated with negative sentiment. This result coupled with the previous result suggest that participants may be eager to please the PM through expressions of positive sentiment and agreement, and more willing to express negative sentiment to non-PM participants.

Q11: Which non-decision states that occur early in meetings tend to cause decisions to be made quickly?

The PRISM property encoding this query is:

```
P=?[F<=50 ((y=j) & "noddecision" & P>=1[X "decision"])]
```

where we considered early meetings to be within 50 time steps. This property computes the probability of eventually (i.e., in the Future) to reach a *noddecision* state identified by *j* within 50 time steps and in the neXt state a *decision* is taken (with probability 1).

The top ten results (states *j* and probabilities) are the following:

StateLabel	Result
PM_inf_nosentiment_noddecision_noaction	0.9677
PM_stl_nosentiment_noddecision_noaction	0.8047
ID_bck_nosentiment_noddecision_noaction	0.6589
UI_stl_nosentiment_noddecision_noaction	0.5329
ID_elInf_nosentiment_noddecision_noaction	0.1977
PM_off_nosentiment_noddecision_noaction	0.1639
ID_und_nosentiment_noddecision_noaction	0.1286
PM_elSug_nosentiment_noddecision_noaction	0.1120
UI_oth_nosentiment_noddecision_noaction	0.1093
ID_off_nosentiment_noddecision_noaction	0.0715

Interestingly, none of these states involve sentiment, and they belong to a variety of the roles. However, the top two results both belong to the PM. This reveals that sentiment and decision-making are less associated with each other early on in the meetings.

Q12: If one person expresses positive sentiment, does it lead to other people expressing positive sentiment?

We compare the average probability of expressing one type of sentiment after another or the same type of sentiment using the following PRISM properties and their results:

Property	Result
filter(avg,P=?[F "positive"],"positive")	0.46
filter(avg,P=?[F "positive"],"negative")	0.19
filter(avg,P=?[F "negative"],"negative")	0.30
filter(avg,P=?[F "negative"],"positive")	0.04

For example, the last property above computes for each *positive* state *s* the probability of reaching a *negative* state when starting from *s*, and then returns the average over all *positive* states *s*.

These results show that an expression of positive sentiment is very likely to be followed by another expression of positive sentiment, and similarly with negative sentiment following negative sentiment. It is less common for negative to follow positive and vice-versa, which is partly reflecting the fact that negative sentiment is much less common in this corpus.

Q13: If a PM person expresses positive sentiments, what is the probability that it leads to positive sentiment expressed by a non-PM person?

This query is a form of causality relation between positive sentiments expressed by a PM person and a non-PM person. We formalise query Q13 as a probabilistic constrained response [9] where we instantiate roleLabel by ME, UI, or ID:

```
P>=1 [G (("PM" & "positive") =>
  P>=p [(!("roleLabel" & "negative") &
    !("PM" & "negative"))
  U<=N ("roleLabel" & "positive")]]]
```

This PRISM property states the following: whenever PM expresses positive sentiment then, with probability greater than *p*, roleLabel and PM do not express negative sentiment until roleLabel expresses a positive sentiment within *N* time steps.

This property helps us identify the maximum probability *p* for which the answer to the query is true when instantiating the roleLabel for non-PM roles. For *N* = 100, then the maximum probabilities *p* for which the answers to Q13 are true are 0.1 for ME, 0.06 for ID, and 0.05 for UI respectively. For *N* = 500, then the

maximum probabilities p for which the answers to Q13 are true are 0.4 for ME, 0.25 for ID, and 0.25 for UI. We conclude that ME is approximately twice as likely than ID and UI to respond positively to a PM positive sentiment.

This result is likely to reflect the structure of the AMI scenario. It tells us that the ME has a great deal of responsibility and can perhaps be seen as a secondary leader of the meeting.

6 CONCLUSION

In this paper we demonstrated the expressiveness of probabilistic temporal logic properties for formalising various probabilistic and reward-based queries about group interactions in meetings and then analysed them with the probabilistic model checker PRISM and interpreted them for the AMI corpus.

Some of the queries analysed above do not need probabilistic temporal logic properties to be asked on the initial data set. However, all queries involving bounded time steps and in particular the steady-state properties, e.g. Q11 and Q13, cannot be expressed in any other way than as temporal property formulae. The queries Q1 - Q6 validate our behavioral model as their results confirm expected interactions, while the queries Q7 - Q13 highlight novel insight into the AMI dataset we analysed.

In this paper we analysed the Markov model inferred from state transitions counts in the data. For future work we will consider admixture models inferred from the data using classical Expectation-Maximisation algorithms where each component (associated with a latent variable) in the admixture model models a particular pattern of behaviour, similar to the work of [2, 3]. The challenge will be in identifying suitable classes of probabilistic temporal properties for characterising and discriminating between the patterns for the particular type of interaction data contained AIM corpus.

In future work, we will experiment with alternative state representations, particularly representations that are less specific to the AMI corpus scenario and its roles. For example, we will include demographic characteristics such as gender and the native language of the speaker. We will also apply this representation and methodology to other group interaction datasets such as the ELEA corpus [17].

Acknowledgement. Oana Andrei's research work has been supported by the EPSRC UK Programme Grant *Science of Sensor Systems Software* (EP/N007565) and the University of Glasgow *John*

Robertson Bequest Fund. Gabriel Murray is supported by an NSERC Discovery grant. The authors also gratefully acknowledge helpful feedback from the anonymous reviewers.

REFERENCES

- [1] J. A. Allen, N. Lehmann-Willenbrock, and S. G. Rogelberg. *The Cambridge handbook of meeting science*. Cambridge University Press, 2015.
- [2] O. Andrei and M. Calder. Data-driven modelling and probabilistic analysis of interactive software usage. *Journal of Algebraic and Logical Methods in Programming (JLAMP)*, 100:195–214, November, 2018.
- [3] O. Andrei, M. Calder, M. Chalmers, A. Morrison, and M. Rost. Probabilistic Formal Analysis of App Usage to Inform Redesign. In *Proceedings of iFM 2016*, volume 9681 of *Lecture Notes in Computer Science*, pages 115–129. Springer, 2016.
- [4] C. Baier and J.-P. Katoen. *Principles of Model Checking*. The MIT Press, 2008.
- [5] J. Carletta. Unleashing the killer corpus: experiences in creating the multi-everything ami meeting corpus. *Language Resources and Evaluation*, 41(2):181–190, 2007.
- [6] C. Cioffi-Revilla. *Introduction to Computational Social Science - Principles and Applications*. Texts in Computer Science. Springer, 2017.
- [7] B. Cornwell. *Social sequence analysis: Methods and applications*, volume 37. Cambridge University Press, 2015.
- [8] A. Dielmann and S. Renals. Recognition of dialogue acts in multiparty meetings using a switching dbn. *IEEE transactions on audio, speech, and language processing*, 16(7):1303–1314, 2008.
- [9] L. Grunske. Specification patterns for probabilistic quality properties. In W. Schäfer, M. B. Dwyer, and V. Gruhn, editors, *Proceedings of ICSE'08*, pages 31–40. ACM, 2008.
- [10] B. Kim and C. Rudin. Learning about meetings. *Data mining and knowledge discovery*, 28(5-6):1134–1157, 2014.
- [11] M. Z. Kwiatkowska, G. Norman, and D. Parker. PRISM 4.0: Verification of Probabilistic Real-Time Systems. In G. Gopalakrishnan and S. Qadeer, editors, *Proceedings of CAV'11*, volume 6806 of *Lecture Notes in Computer Science*, pages 585–591. Springer, 2011.
- [12] D. Lazer, A. Pentland, L. Adamic, S. Aral, A.-L. Barabási, D. Brewer, N. Christakis, N. Contractor, J. Fowler, M. Gutmann, T. Jebara, G. King, M. Macy, D. Roy, and M. Van Alstyne. Computational Social Science. *Science*, 323(5915):721–723, 2009.
- [13] G. Murray. Markov reward models for analyzing group interaction. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction*, pages 336–340. ACM, 2017.
- [14] G. Murray. Modelling participation in small group social sequences with markov rewards analysis. In *Proceedings of the Second Workshop on NLP and Computational Social Science*, pages 68–72, 2017.
- [15] A. Pilny and M. S. Poole, editors. *Group Processes - Data-Driven Computational Approaches*. Computational Social Sciences. Springer, 2017.
- [16] S. Renals, H. Bourlard, J. Carletta, and A. Popescu-Belis. *Multimodal Signal Processing: Human Interactions in Meetings*. Cambridge University Press, 2012.
- [17] D. Sanchez-Cortes, O. Aran, M. S. Mast, and D. Gatica-Perez. A nonverbal behavior approach to identify emergent leaders in small groups. *IEEE Transactions on Multimedia*, 14(3):816–832, 2012.
- [18] A. Vinciarelli, M. Pantic, and H. Bourlard. Social signal processing: Survey of an emerging domain. *Image and vision computing*, 27(12):1743–1759, 2009.