Markov Reward Models for Analyzing Group Interaction

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

ABSTRACT

In this work we introduce a novel application of Markov Reward models for studying group interaction. We describe a sample state representation for social sequences in meetings, and give examples of how particular states can be associated with immediate positive or negative rewards, based on outcomes of interest. We then present a Value Iteration algorithm for estimating the values of states. While we focus on two specific applications of Markov Reward models to small group interaction in meetings, there are many ways in which such a model can be used to study different facets of group dynamics and interaction. To encourage such research, we are making the Value Iteration software freely available.

CCS CONCEPTS

• **Computing methodologies → Discourse, dialogue and pragmatics**; Markov decision processes;

KEYWORDS

Small groups, social sequences, markov rewards, discourse structure, meetings

ACM Reference Format:
Gabriel Murray. 2017. Markov Reward Models for Analyzing Group Interaction. In *Proceedings of 19th ACM International Conference on Multimodal Interaction (ICMI'17).* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3136755.3136778

1 INTRODUCTION

Researchers from numerous fields are interested in better understanding small group dynamics and group interaction. Analyzing group interactions can help shed light on sociological and social psychological questions relating to group network structure, evolving ties between members, participant dominance, social influence, and many other phenomena. As datasets of group interactions have become more readily available, data-driven computational approaches to group analysis have come to the fore [14]. Algorithms that are capable of processing large amounts of group interaction data to shed light on interesting patterns and relationships are particularly useful in this resurgent field of research. In turn, such computational approaches can be incorporated into systems that provide feedback to group members in order to facilitate group interaction in meetings [17, 21].

In this paper we describe a novel application of Markov Reward models for studying group interaction. We describe a sample state representation for social sequences in small group meetings. Particular sequence states can be associated with positive or negative rewards, depending on the outcomes of interest that are being analyzed. We provide two examples of such outcomes and their associated state rewards. We then present a Value Iteration algorithm that allows us to estimate the value of every state, given the state transitions and the specified rewarding states. We are making the Value Iteration software freely available to encourage similar research on other outcomes of interest.

One strength of the Markov Reward model for analyzing group interaction is that it can identify relationships between sequence states that are not temporally adjacent. A second strength is that it is easily tunable to allow the experimenter to increase or decrease the impact that more distant states have on each other. Third, this approach is very flexible in terms of state representations and reward configurations that are possible.

In Section 2, we survey a wide variety of tools that have been used for studying group interaction. In Section 3, we present our Markov Reward model, including the sample state representation and a Value Iteration algorithm. Section 4 describes the meeting corpus used for these experiments. We describe the experiments and results in Section 5 and conclude in Section 6.

2 RELATED WORK

We briefly consider a variety of tools that have been used in recent research in order to understand different aspects of group interaction.

Recurrence Analysis. With recurrence analysis [9], the goal is to analyze the prevalence of recurring patterns in group interaction. Recurrence analysis of a small group interaction involves the creation of a recurrence plot containing information about sequence states being revisited. Various statistics can then be calculated from the plot; for example, the *recurrence rate* is the density of recurrences in the plot [9].

Sequential Analysis. With sequential analysis [8], the goal is to look for any association between certain social behaviours and subsequent behaviours. One method for doing this is to analyze the interaction using different time-lags. For example, Klonek et al. [8] create matrices containing information about the frequency of sequentially adjacent behaviours, and then conduct statistical analyses of the matrices. In contrast, the method we describe in this paper has the benefit of detecting associations between states that may not be adjacent, by employing a Value Iteration algorithm.

ICMI'17, November 13–17, 2017, Glasgow, UK

ACM ISBN 978-1-4503-5543-8/17/11...$15.00
https://doi.org/10.1145/3136755.3136778

Social Sequence Analysis. Social sequence analysis [5] comprises a set of tools and techniques that sociologists use to study social sequence data, where the sequences are usually temporal and often unfold over days, months or years, rather than minutes or hours. One such technique is *optimal matching* [5], where social sequences are compared using edit distance metrics that have been borrowed and adapted from the field of bioinformatics. Such analysis may also involve social network structure [6] and the study of how networks evolve over time. Relational Event Models [3, 15] combine aspects of social sequence analysis and social network analysis.

Machine Learning. Machine learning and natural language processing have been used in recent research on meeting interactions, both for prediction and inference purposes. For example, machine learning models can predict whether the group is currently making a decision [7] or discussing an action item [13, 16]. It has also been used to analyze the ways in which productive and unproductive meetings differ in terms of their features [12].

Social Signal Processing. Many different tools and techniques have been used to analyze multi-modal aspects of small group interaction. These include gesture recognition, voice recognition, dialogue act detection, meeting summarization, face detection, and sentiment detection [2, 17, 22]. Most of these systems rely on machine learning algorithms.

Applications of Markov Reward Models. Finally, we note that Markov Reward models have been utilized in many contexts, from estimating the cost of geriatric care [10] to determining the values of various actions in sports [11, 18]. They have also been widely used in performability and reliability analyses [20]. To our knowledge, they have not previously been used for studying and understanding group dynamics and small group interaction. This type of reward model and associated algorithm are closely related to Value Iteration for Markov Decision Processes (MDPs), and MDPs have long been used in dialogue systems [23].

3 MARKOV REWARD MODELS

In this section we describe the state representation used in our Markov Reward models, and the Value Iteration algorithm used for estimating the values of each state.

3.1 State Representation

There are many possible state representations for group interaction in meetings. For these experiments, we use complex states representations for the social sequences, where each state is a 4-tuple consisting of the following information:

- the participant's role in the group
- the dialogue act type
- the sentiment being expressed (positive, negative, both, none)
- whether the utterance involves a decision

In social sequence analysis, using complex state representations in this manner is called *alphabet expansion* [5], and contrasts with sequential analysis of a single dimension or single phenomenon.

For the corpus we use (Section 4), the participant roles are precisely defined: Project Manager (PM), Marketing Expert (ME), User Interface Designer (UI), and Industrial Designer (ID).

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

We utilize the AMI dialogue act annotation scheme [17] for the dialogue act type information, summarized in Table 1. Longer descriptions and further information on the AMI meeting annotations can be found in Renals et al. [17].

Example states include the following:

- $< PM - bck - pos - nodec >$ (the project manager making a positive back-channel comment, unrelated to a decision)
- $< PM - el.ass - nosent - yesdec >$ (the project manager eliciting feedback about a decision item)
- $< UI - sug - nosent - yesdec >$ (the UI expert making a suggestion about a decision item)

3.2 Value Iteration

Given the state representation just defined, particular states can be associated with positive or negative rewards. In Section 5 we will give two examples of associating particular states with rewards, based on outcomes of interest. Once we have associated certain states with immediate rewards, we can then use the Value Iteration algorithm to determine the estimated value of every state. For example, if all decision states have an immediate reward of 1, it may be the case that there is some non-decision state that has a high estimated value because it tends to lead to decision items. This highlights the difference between an immediate reward of a state and the estimated value of a state, where the latter is based on which other states you can transition to, and what the values of those other states are.

The state transition probabilities are estimated directly from the data. The Markov assumption in the Markov Rewards model is that the probability of a given state depends only on the preceding state in the sequence. In addition to the complex states described in the preceding section, there are START and STOP states representing the beginning and end of a meeting, and the STOP state is absorbing, i.e. there are no transitions out of the STOP state.

Algorithm 1 shows the Value Iteration algorithm for our Markov Rewards model. The inputs are an initial reward vector r containing the immediate rewards for each state, a transition matrix M, and a discount factor γ. The algorithm outputs a vector v containing the estimated values of each state. The core of the algorithm is an update equation that is applied until convergence, when the elements of v are no longer changing, or changing by only a very small amount.

Algorithm 1: Value Iteration for Markov Rewards Model

Input: reward vector r, transition matrix M, discount factor γ
Output: A vector v containing the estimated values of all states
$v_0 = r$
$t = 1$
repeat
  $v_t = r + \gamma \cdot (M \cdot v_{t-1})$
  $t = t + 1$
until *convergence*;
return v_{t-1}

The update equation $v_t = r + \gamma \cdot (M \cdot v_{t-1})$ essentially says that the states at step t of the algorithm have an estimated value equal to their immediate reward, plus the discounted value – calculated at the previous step $t - 1$ – of the states that can be transitioned to. Intuitively, the estimated value of state i at time step t of the algorithm, indicated by v_t^i, indicates the expected value of being in that state if there are t steps left in the social sequence. When Value Iteration converges, the final estimate v^i indicates the estimated long-term value of being in state i.

The discount factor γ can be set to a value between 0 and 1, and controls how much weight is given to future rewards, compared with immediate rewards. For example, $\gamma = 0$ would give no weight to future rewards. A setting greater than 0 and less than 1 means that rewards have less weight the further away in the sequence they are. An advantage of the γ parameter then is that it allows the experimenter to control how much states can affect other distant states. For our experiments, we set $\gamma = 0.9$.

Convergence is reached when there is an iteration after which no state value has changed from its previous value by more than some threshold τ. In this work, we set $\tau = 0.001$. Convergence is typically reached in fewer than 60 iterations.

We have made the software for running Value Iteration, and replicating these results, freely available[1].

4 CORPUS

For this study, we use the AMI meeting corpus [4], a corpus of scenario and non-scenario meetings. In the scenario subset of the corpus, each meeting consists of four participants who are role-playing as members of a company tasked with designing a remote control unit. The participants are assigned the roles mentioned previously: project manager (PM), user interface expert (UI), marketing expert (ME), and industrial designer (ID). While the scenario given to each team is artificial and structured, the participation and interaction of the group members is not scripted. The conversation is natural and spontaneous, and the groups can make whatever decisions they see fit.

For these experiments, we rely on the AMI gold-standard annotations for dialogue act type, sentiment type, and decision items [17]. Because our state representation utilizes sentiment information, and sentiment is also one of our outcomes of interest in the experiments below, we focus on a subset of 20 AMI meetings that have been annotated with sentiment information [24].

Table 2: Estimated State Values w/ Decision Rewards

State	Val.	Freq.
$< PM - stl - pos - nodec >$	0.192	74
$< ME - stl - neg - nodec >$	0.184	17
$< PM - fra - pos - nodec >$	0.177	27
$< ID - bck - pos - nodec >$	0.173	31
$< ID - stl - pos - nodec >$	0.168	24
$< PM - stl - neg - nodec >$	0.141	19
$< UI - el.sug - nosent - nodec >$	0.139	18
$< PM - inf - neg - nodec >$	0.122	113
$< UI - el.ass - nosent - nodec >$	0.121	62
$< PM - el.sug - nosent - nodec >$	0.120	68
$< ME - off - nosent - nodec >$	0.086	45
$< ID - sug - neg - nodec >$	0.086	15
$< ID - el.inf - neg - nodec >$	0.085	19
$< ID - oth - pos - nodec >$	0.085	12
$< UI - fra - neg - nodec >$	0.085	17
$< ME - be.pos - nosent - nodec >$	0.085	46
$< UI - el.inf - pos - nodec >$	0.084	10
$< ID - el.und - nosent - nodec >$	0.082	6
$< UI - stl - neg - nodec >$	0.081	14
$< ID - stl - neg - nodec >$	0.080	12

5 EXPERIMENTS AND RESULTS

In the following subsections we describe two sets of experiments, based on two different outcomes of interest. In both cases the state representations and Value Iteration algorithm are the same, and only the state rewards differ.

5.1 Experiment I: Decision Rewards

For our first set of experiments, the rewarding states are any states that include a decision being made. That is, states consisting of a four-tuple $< A, B, C, yesdec >$, where A, B, and C can take on any values. These states are of interest because the point of meetings is often to come to some type of group decision, and it can be helpful to know what types of states tend to precede decisions. All decision states are assigned an immediate reward of 1, and all other states have rewards of 0.

Table 2 shows the estimated values after running Value Iteration, for selected states. Specifically, it shows the estimated values of the top 10 and bottom 10 non-decision states, i.e. states that do not have immediate rewards. We can notice that most of the top 10

[1] https://github.com/gmfraser/markov-rewards

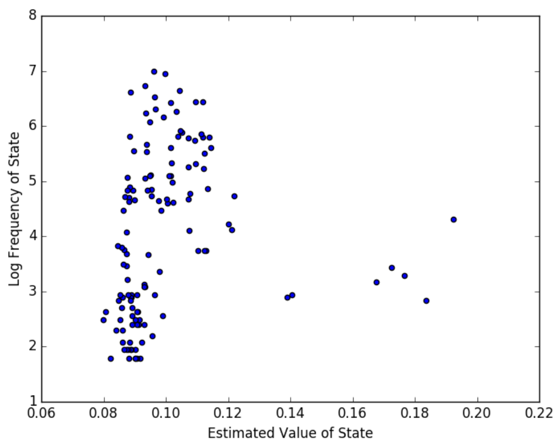

Figure 1: State Frequency vs. Value, w/ Decision Rewards

states associated with decisions contain sentiment, either positive or negative, and often belong to the project manager. In contrast, the bottom 10 states mostly contain either negative sentiment or no sentiment and belong to group members other than the project manager. Of the top 10 states listed, the most frequent in the corpus are $< PM - stl - pos - nodec >$ and $< PM - inf - neg - nodec >$. Both involve the project managers expressing sentiment, illustrating that team leaders tend to express their own positive and negative opinions as the group begins to make a decision.

For each state type, Figure 1 shows its estimated value versus its log-frequency in the corpus. We can see that a fairly small number of medium-frequency states have the highest estimated values.

5.2 Experiment II: Sentiment Rewards

For our second set of experiments, the rewarding states are those states containing sentiment. Positive sentiment states have a reward of 1, while negative sentiment states have a reward of -2. This is in line with previous work that has found that negative sentiment carries more *cognitive weight* than positive sentiment [19], and that language tends to have a positive bias [1, 19].

Once again we run the Value Iteration algorithm and report the top 10 and bottom 10 non-sentiment states, with the results shown in Table 3. Notice that even the bottom 10 states have positive scores; this is owing to the fact that positive sentiment is much more prevalent than negative sentiment in this corpus (and in language in general, as stated earlier). We note that the top and bottom states often contain decisions, indicating that decisions or decision proposals frequently lead to statements containing positive or negative sentiment. We also see that the top two states belong to the project manager, as well as the most frequent negative state, $< PM - ass - nosent - nodec >$. Many of the top 10 and bottom 10 states involve a group member trying to elicit input from the other members. For example, $< ME - el.ass - nosent - nodec >$ is the most frequent of the positive states.

Table 3: Estimated State Values w/ Sentiment Rewards

State	Val.	Freq.
$< PM - el.ass - nosent - yesdec >$	0.943	8
$< PM - sug - nosent - yesdec >$	0.939	8
$< ME - be.neg - nosent - nodec >$	0.754	6
$< ID - el.sug - nosent - nodec >$	0.693	18
$< UI - off - nosent - nodec >$	0.693	32
$< ID - el.und - nosent - nodec >$	0.681	6
$< ME - el.sug - nosent - nodec >$	0.658	25
$< ME - el.ass - nosent - nodec >$	0.635	105
$< ID - inf - nosent - yesdec >$	0.631	11
$< PM - el.und - nosent - nodec >$	0.623	14
$< ID - sug - nosent - nodec >$	0.473	158
$< PM - oth - nosent - nodec >$	0.473	119
$< ME - off - nosent - nodec >$	0.472	45
$< PM - ass - nosent - nodec >$	0.465	513
$< UI - el.sug - nosent - nodec >$	0.463	18
$< ME - el.und - nosent - nodec >$	0.426	18
$< ID - el.ass - nosent - nodec >$	0.425	61
$< UI - sug - nosent - nodec >$	0.405	103
$< UI - inf - nosent - yesdec >$	0.216	8
$< ME - inf - nosent - yesdec >$	0.131	6

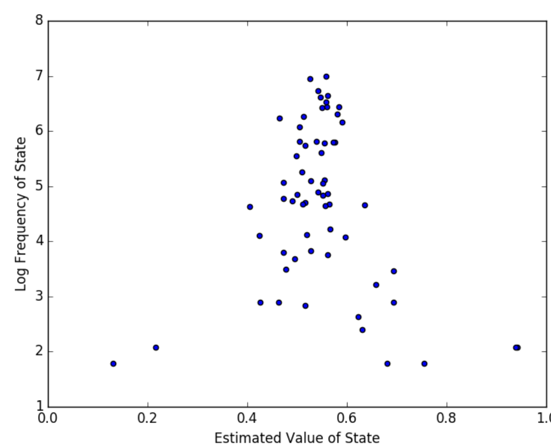

Figure 2: State Frequency vs. Value, w/ Sentiment Rewards

For each state type, Figure 2 shows its estimated value versus its log-frequency. Again we see that relatively few mid-frequency states have the highest estimated values.

6 CONCLUSION

We have introduced a novel application of Markov Reward models for understanding small group interaction. We presented a sample state representation for meeting interactions, which combines information about dialogue act types, participant roles, and discourse information. Many other state representations are possible. We also demonstrated how particular states can be associated with positive or negative rewards, using two outcomes of interest as examples: decision items, and sentiment items. We also presented a Value Iteration algorithm that allows us to estimate the value of every state.

Markov Reward models have great strengths and flexibility for studying group interaction. They can detect relationships between non-adjacent states, are adjustable in terms of being able to increase or decrease the impact that distant states have on each other, and are very flexible in terms of state representations and reward configurations that are permitted. To encourage similar analysis with other outcomes of interest, different state representations, and varying system parameters, we are making the Value Iteration software freely available.

REFERENCES

[1] J. Boucher and C. Osgood. 1969. The pollyanna hypothesis. *Journal of Verbal Learning and Verbal Behavior* 8, 1 (1969), 1–8.

[2] J. Burgoon, N. Magnenat-Thalmann, M. Pantic, and A. Vinciarelli. 2017. *Social Signal Processing.* Cambridge University Press.

[3] C. Butts. 2008. A relational event framework for social action. *Sociological Methodology* 38, 1 (2008), 155–200.

[4] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, and M. Kronenthal et al. 2005. The AMI meeting corpus: A pre-announcement. In *International Workshop on Machine Learning for Multimodal Interaction.* Springer, 28–39.

[5] B. Cornwell. 2015. *Social sequence analysis: Methods and applications.* Vol. 37. Cambridge University Press.

[6] N. Friedkin and E. Johnsen. 2011. *Social influence network theory: A sociological examination of small group dynamics.* Vol. 33. Cambridge University Press.

[7] B. Kim and C. Rudin. 2014. Learning about meetings. *Data Mining and Knowledge Discovery* 28, 5-6 (2014), 1134–1157.

[8] F. Klonek, V. Quera, M. Burba, and S. Kauffeld. 2016. Group interactions and time: Using sequential analysis to study group dynamics in project meetings. *Group Dynamics: Theory, Research, and Practice* 20, 3 (2016), 209–222.

[9] A. Knight, D. Kennedy, and S. McComb. 2016. Using recurrence analysis to examine group dynamics. *Group Dynamics: Theory, Research, and Practice* 20, 3 (2016), 223–241.

[10] S.I. McClean, B. McAlea, and P.H. Millard. 1998. Using a Markov reward model to estimate spend-down costs for a geriatric department. *Journal of the Operational Research Society* 49, 10 (1998), 1021–1025.

[11] M. Miskin, G. Fellingham, and L. Florence. 2010. Skill Importance in Women's Volleyball. *Journal of Quantitative Analysis in Sports* 6, 2 (2010).

[12] G. Murray. 2014. Learning how productive and unproductive meetings differ. In *Canadian Conference on Artificial Intelligence.* Springer, 191–202.

[13] G. Murray and S. Renals. 2008. Detecting Action Items in Meetings. In *Proc. of MLMI 2008, Utrecht, the Netherlands.*

[14] A. Pilny and M. Poole. 2017. *Group Processes: Data-Driven Computational Approaches.* Springer.

[15] A. Pilny, A. Schecter, M. Poole, and N. Contractor. 2016. An illustration of the relational event model to analyze group interaction processes. *Group Dynamics: Theory, Research, and Practice* 20, 3 (2016), 181–195.

[16] M. Purver, J. Dowding, J. Niekrasz, P. Ehlen, and S. Noorbaloochi. 2007. Detecting and Summarizing Action Items in Multi-Party Dialogue. In *Proc. of the 9th SIGdial Workshop on Discourse and Dialogue, Antwerp, Belgium.*

[17] S. Renals, H. Bourlard, J. Carletta, and A. Popescu-Bellis. 2012. *Multimodal Signal Processing: Human Interactions in Meetings.* Cambridge University Press.

[18] K. Routley and O. Schulte. 2015. A Markov Game model for valuing player actions in ice hockey. In *Proc. of UAI.* AUAI Press, 782–791.

[19] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede. 2011. Lexicon-based Methods for Sentiment Analysis. *Computational Linguistics* 37, 2 (June 2011), 267–307. https://doi.org/10.1162/COLI_a_00049

[20] K. Trivedi, M. Malhotra, and R. Fricks. 1994. Markov reward approach to performability and reliability analysis. In *Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 1994., MASCOTS'94., Proceedings of the Second International Workshop on.* IEEE, 7–11.

[21] G. Tur, A. Stolcke, L. Voss, et al. 2008. The CALO meeting speech recognition and understanding system. In *Spoken Language Technology Workshop, 2008. SLT 2008. IEEE.* IEEE, 69–72.

[22] A. Vinciarelli, M. Pantic, and H. Bourlard. 2009. Social signal processing: Survey of an emerging domain. *Image and vision computing* 27, 12 (2009), 1743–1759.

[23] J.D. Williams and S. Young. 2007. Partially observable Markov decision processes for spoken dialog systems. *Computer Speech & Language* 21, 2 (2007), 393–422.

[24] T. Wilson. 2008. Annotating Subjective Content in Meetings. In *Proc. of LREC 2008, Marrakech, Morocco.*