An experimental analysis of whispers' effect in Werewolf BBS by relational association rules

Saki Sakaguchi and Tomonobu Ozaki

Graduate School of Integrated Basic Sciences, Nihon University
3-25-40 Sakurajosui, Setagaya-ku, Tokyo 156-8550, Japan
saki.sakaguchi@gmail.com, tozaki@chs.nihon-u.ac.jp

Abstract. The Werewolf game is a conversation-based party games. Each player in the game belongs to werewolves or villagers. Since secret conversations called "Whispers" are allowed for werewolves only, effective use of whispers must be a key issue for werewolves to proceed advantageously to win the game. In this work-in-progress paper, for a preliminary assessment of the whispers' effect, we extract relational association rules having behaviors in whisper from the log data of Werewolf BBS.

Keywords: Werewolf game, relational association rules, log analysis

1 Introduction

In recent years, there has been a growing interest in the research of Artificial Intelligence. Current technologies in AI reach a level high enough to beat human in complete information games such as Shogi and Go. As a next step for realizing general artificial intelligence, incomplete information games are receiving increased attention. As one of representative incomplete information games, the Werewolf game is widely recognized as promising research testbed for intelligent agents in Japan, and a project for making artificial intelligence based Werewolf (AIWolf)[1] is established recently. Intensive researches are conducted from various aspects for realizing AIWolf, *e.g.* [**?**,**?**].

The Werewolf game is a conversation-based party game which models a conflict between werewolves who are minorities having rich information and villagers who are majorities having less information. There exist two types of conversations in the game. One is an open conversation, and the other is a closed or secret conversation. While all alive players in the game can join and browse the open conversations, secret conversations are allowed for werewolves only. Thus, effective use of secret conversations must be a key issue for werewolves to proceed advantageously to win the game.

In this work-in-progress paper, we focus on information differences between werewolves and villagers, and try to capture a characteristic relationship between

[1] http://aiwolf.org/en/

contents in the secret conversations and actual utterances in the open conversations. For this purpose, we extract relational association rules[**?**,**?**] having high confidence value whose head is an utterance in open conversations and whose body has at least one contents in the secret ones.

2 Modeling the Werewolf games in Logic

2.1 The Werewolf game

The Werewolf game is a multiplayer communication party game. Each player belongs to werewolves side or villagers side. A werewolf player knows who belong to the same side, but villagers have no information on other players' side. Some villager has a special ability. Seers can know that the designated player is a werewolf or not. Mediums can know that an executed player was a werewolf. Hunter can guard a designated player from the attack by werewolves. The game has two phases, daytime phase and nighttime phase, to be iterated. In daytime phases, all players join the open conversation and give vote for deciding an executed player. In the conversation, villagers try to find out werewolves and werewolves try to deceive villagers. In nighttime phase, werewolves select a dangerous villager and attack him/her. Executed or attacked players are exiled from the game. Villagers win the game if all werewolves are executed, while werewolves win if the number of villagers is no more than that of werewolves.

The Werewolf BBS[2] is an online BBS website for playing text-based Werewolf games. The rules in the BBS are almost the same as those in the original Werewolf games with a few exceptions. The BBS has four types of log data storing players' utterances. A "white log" stores all utterances during the open conversations. All players can browse a white log. A "red log" keeps the utterances called "whispers" in a secret conversation among werewolves. We employ these two kinds of log data for the analysis.

2.2 Predicates for representing utterances

Each utterance is written in natural language. To extract essential meanings of utterances and convert them machine manageable, a communication protocol for the Werewolf game is proposed in [**?**]. By using the communication protocol as a reference, we prepare fourteen predicates for representing a meaning of utterances in the white log as well as thirteen ones for whispers. Hereafter, for the simplicity, we call predicates for the white and red logs as "white predicate" and "red predicate", respectively. A few examples of white predicates are explained below.

`w_question( Game:Day, Player, Player2 )` : A player `Player` asks a player `Player2` a question on the `Day`th day in a game `Game`.

[2] `http://www.wolfg.x0.com/`

`w_request_divine( Game:Day, Player, Player2 )` : A player `Player` requests seers to divine the team which a player `Player2` belongs to on the `Day`th day in a game `Game`.

A complete list of red predicates is shown in Table **??**. We explain a couple of red predicates below.

`r_want_eat( Game:Day, Player, Players2 )` : A werewolf `Player` wants to attack a player `Player2` on the `Day`th day in a game `Game`.

`r_estimate( Game:Day, Players, Player2, Role)` : A werewolf `Player` estimates that a player `Player2` has a role of `Role` on the `Day`th day in a game `Game`.

`r_decieve( Game:Day, Player, Player2, Role)` : A werewolf `Player` offers a werewolf `Player2` to behave as `Role` to deceive villagers on the `Day`th day in a game `Game`.

Three arguments, `Game`, `Day` and `Player` are in common in all predicates for handling a chain of utterances. In addition, to relate the past utterances to the current one, a rule

Pred`( Game:Day, N, Player,` *Args*$\cdots$ `) :-`
`prev_days(N), PDay is Day-N,` *Pred*`( Game:PDay, Player,` *Args*$\cdots$`).`

is employed for each predicate, in which a predicate `prev_days(N)` returns a non-negative integer `N`. This rule states that a player `Player` took an action *Pred* `N` days ago from `Day`th day in a game `Game`.

3 Mining relational association rules

3.1 Dataset

We select six games from the Werewolf BBS. All of them have twelve villagers and three werewolves including at least deceiving one. Werewolves won three of six games, and lost the rest three.

All white and red logs are manually converted into the predicates introduced in the previous section. The average numbers of facts on red predicates over three games the werewolves won and lost respectively are summarized in Table **??**. From the table, we can confirm that the main topics in secret conversations are question, answer, advice, estimate, and want_eat. Furthermore, each number in the games werewolves won is more than that in the games werewolves lost, even if we consider the number of werewolves executed. In other words, intensive communications are observed in the game werewolves won.

3.2 Restriction and evaluation measure

In this work-in-progress paper, relational association rules to be extracted are restricted to have at least one red predicate in their body. Furthermore, they have to contain one of three head predicates below:

Table 1. The average numbers of facts on red predicates per day

Games Werewolves won									Games Werewolves lost						
1st	2nd	3rd	4th	5th	6th	7th	8th	red predicate	1st	2nd	3rd	4th	5th	6th	7th
6.7	10.0	6.0	6.3	6.7	5.3	2.7	–	r_question	7.7	5.0	7.3	2.3	1.7	2.0	–
6.7	7.7	6.0	5.0	4.0	5.0	2.3	–	r_answer	7.7	5.3	6.3	1.3	0.3	1.3	–
3.3	6.3	2.0	2.7	4.0	5.0	1.3	–	r_adviced	4.7	3.0	5.0	1.7	–	0.7	–
2.7	7.0	4.7	7.7	1.7	2.3	1.7	–	r_estimate	7.3	4.3	2.3	1.7	2.3	–	–
2.3	1.0	4.0	0.7	0.7	1.7	0.3	0.3	r_agree	0.7	–	–	–	–	–	–
–	–	0.3	–	0.3	0.3	–	–	r_disagree	–	0.3	–	0.3	–	–	–
3.3	5.7	6.7	4.0	4.7	3.7	2.7	0.7	r_want_eat	0.7	3.0	7.7	5.0	2.3	2.3	–
–	0.3	–	0.3	–	0.3	–	1.0	r_want_vote	0.7	1.0	0.3	0.7	0.3	–	–
–	–	1.0	–	1.0	0.7	–	–	r_black_paint	0.3	0.3	1.0	1.0	0.3	2.0	–
–	0.3	–	1.0	–	–	0.3	–	r_disrelation	0.3	–	2.3	0.3	–	–	–
1.3	–	–	–	–	0.7	0.7	–	r_deceive	1.7	–	–	0.3	–	–	–
0.3	–	–	–	–	–	–	–	r_hide	0.7	–	–	–	–	–	–
9.0	9.0	8.0	8.0	8.0	7.0	5.0	4.0	r_say_count	9.0	9.0	9.0	8.0	6.0	5.0	3.0

`attacked( Game:Day, Player)` : A player `Player` was attacked by werewolves on the `Day`th day in a game `Game`.

`executed( Game:Day, Player)` : A player `Player` was executed by the vote on the `Day`th day in a game `Game`.

`wolves_estimate_wolf( Game:Day, Werewolf, Player)` : A werewolf `Werewolf` sayed that a player `Player` is a werewolf on the `Day`th day in a game `Game`.

Note that, since werewolves know who werewolves are, designating villager as a werewolf in the predicate `wolves_estimate_wolf` indicate that a werewolf tries to deceive other villagers. Werewolves may also designate a werewolf to avoid a suspicion. We extract facts on the above three head predicates from log data. As a result, 27, 39 and 53 facts are obtained for `attacked`, `executed` and `wolves_estimate_wolf`, respectively.

Three interestingness measures are used for evaluating relational association rules. The first one is support count which is defined as a number of distinct instantiations of head variables by which we can derive both of head and body. The second one is confidence value or conditional probability. It is defined as a probability that an instantiation of head variables satisfying the body can derive the head. To assess the rough effects of the red predicates in the whole, we employ the third measure $D = P(Head \mid Body) - P(Head)$ where $P(Head \mid Body)$ is the confidence value and $P(Head)$ is a priori probability that the head holds. The value of $P(Head)$ is estimated by using all possible instantiations of head predicate considering alive players and their roles. The positive value of this measure indicates that the body predicates have positive effects to the head, while negative one shows the negative effect of the body.

3.3 Results

An inductive logic programming engine Aleph[3] is employed to extract all association rules regardless of that they contain red predicates or not. We give Aleph system a certain parameter setting for association rule search and execute it with the induce_max command. Relational association rules satisfying our conditions are extracted from the results of Aleph system in a post-processing. As a result, 4702, 3048 and 3094 rules are obtained having the predicate `attacked`, `executed` and `wolves_estimate_wolf`, respectively.

A couple of derived association rules having high confidence value are shown below.

1. Werewolves attack a player `C` whom werewolves want to attack if `C` asked a question for a player `E` estimating a relationship between two players.
```
attacked( Game:Day, C ) :-
  r_want_eat( Game:Day, 0, D, C ),
  w_question( Game:Day, 1, C, E ), w_line( Game:Day, 1, E, F, G ).
```

2. A player `C` is executed if `C` is given a vote by a player `D` whom the werewolves estimated as hunter.
```
executed( Game:Day, C ) :-
    w_vote( Game:Day, 0, D, C ),
    r_estimate( Game:Day, 2, F, D, hunter ).
```

3. A werewolf state that a player `C` is a wolf if a player `E` whom werewolf `F` wants to attack agree with `C`.
```
w_estimate_wolf( Game:Day, C ) :-
    w_agree( Game:Day, 2, E, C),
    r_want_eat( Game:Day, 2, F, E ).
```

Table **??** summarizes how many rules having each red predicate have the positive or negative effects. No rules having `r_disagree`, `r_black_paint` and `r_hide` are extracted. Two red predicates `r_want_eat` and `r_estimate` appear frequently regardless of the head predicates. Most of the red predicates have positive effect without a few exceptions. Three predicates `r_deceive`, `r_disrelation` and `r_agree` appear for `attacked` only with complete positive effects. These results show the intensive discussion among werewolves. The predicate `r_want_vote` tends to have a positive effect for `wolves_estimate_wolf`. This result suggests that a werewolf prompts other players to vote a target player by saying he/she is a werewolf in the open conversation.

4 Conclusion

In this work-in-progress paper, we extract relational association rules which relate secret conversations to real actions from the Werewolf BBS.

[3] `http://www.cs.ox.ac.uk/activities/machinelearning/Aleph/aleph`

Table 2. Numbers of extracted relational association rules having red predicates

predicate	executed			attacked			wolves_estimate_wolf		
	$D > 0$	$D \leq 0$	R*	$D > 0$	$D \leq 0$	R*	$D > 0$	$D \leq 0$	R*
r_question	29	1	0.97	423	25	0.94	30	19	0.61
r_answer	28	6	0.82	545	30	0.95	18	24	0.43
r_adviced	11	2	0.85	344	18	0.95	29	4	0.88
r_estimate	404	155	0.72	465	191	0.71	682	673	0.50
r_agree	–	–	–	87	–	1.00	–	–	–
r_want_eat	1876	492	0.79	3394	568	0.86	413	1241	0.25
r_want_vote	98	23	0.81	64	39	0.62	67	20	0.77
r_disrelation	–	–	–	3	–	1.00	–	–	–
r_deceive	–	–	–	15	–	1.00	–	–	–
r_say_count	14	–	1.00	278	3	0.99	15	–	1.00

* : ratio of $D > 0$

As one of future works, we plan to extract condensed representations of relational association rules[**?**] and evaluate them using various interesting measures. In addition, as one of promising research directions for the assessment of whispers' effect, we investigate propensity score matching[**?**] for relational data[**?**].

Acknowledgements We heartily thank Mr. Ninjin for allowing us to use the log data in the Werewolf BBS. We have deep regards to Professor Fujio Toriumi at the University of Tokyo for providing us the Werewolf databases. A part of this work was supported by JSPS KAKENHI Grant Number JP26330262.

References

1. Fujio Toriumi, Kengo Kajiwara, Hirotaka Osawa, Michimasa Inaba, Daisuke Katagami and Kosuke Shinoda: Development of AI Wolf Server, *The 19th Game Programming Workshop*, 2014 (in Japanese)
2. Hirotaka Osawa: Communication Protocol for the "Werewolf" game, *Human-Agent Interaction Symposium*, 2013 (in Japanese)
3. Luc Dehaspe: Frequent Pattern Discovery in First-Order Logic, Ph.D. dissertation, Katholieke Universiteit Leuven, 1998
4. Luc Dehaspe and Hannu Toivonen: Discovery of frequent DATALOG patterns, *Data Mining and Knowledge Discovery*, Vol.3, Issue 1, pp.7–36, 1999
5. Luc De Raedt and Jan Ramon: Condensed Representations for Inductive Logic Programming, *The 9th International Conference on Principles and Practice of Knowledge Representation*, pp.438–446, 2004
6. Paul. R. Rosenbaum and Donald. B. Rubin: The central role of the propensity score in observational studies for causal effects, *Biometrika*, Vol.70, Issue 1, pp.41-55, 1983
7. David Arbour, Katerina Marazopoulou, Dan Garant and David Jensen: Propensity Score Matching for Causal Inference with Relational Data, *The UAI 2014 Workshop Proceedings of the UAI 2014 Workshop Causal Inference: Learning and Prediction*, pp.25–34, 2014