Enhancing Predictive Models in Persuasion Games: Advanced MAB Simulations

Michal Barsht
Technion
michalbar@campus.technion.ac.il

Amir Steiner
Technion
Amirsteiner@campus.technion.ac.il

Abstract

In this project, we will try to advance the research in predicting binary actions of players in language-based persuasion games by improving the data simulation. First, we propose to enrich the data production phase by incorporating an additional strategy into the simulation process. Our new strategy aims to generate more diversified and realistic scenarios that better reflect human decision-making dynamics, thus enhancing the robustness and applicability of our predictive models.

Second, we intend to refine the strategy selection and enhance the learning process to resemble real human desition makers. By implementing a Multi-Armed Bandit algorithm into the strategy selection we are trying to get more realistic simulated data.

1 Introdaction

Recent advances in Large Language Models (LLMs) have spurred interest in designing LLM-based agents for tasks that involve interaction with human and artificial agents. One key aspect of designing such agents is predicting human decisions in off-policy evaluation (OPE). This task is particularly challenging in language-based persuasion games, where the agent's goal is to influence its partner's decisions through verbal messages.

In this work, we extend the previous research on predicting human choices in language-based persuasion games by integrating a Multi-Armed Bandit (MAB) approach for strategy selection. Our hypothesis is that dynamically selecting strategies based on observed human behavior will improve the accuracy of predictions and overall agent performance.

[1]

The contributions of this paper are as follows: First, we introduce a new approach to strategy selection in human-agent interaction using MAB algorithm. Then, we demonstrate a new strategy for simulating decision-making actions.

2 Related Work

The field of predicting human decisions in interactive settings has seen substantial advancements, particularly with the advent of LLMs. Researchers have explored various approaches to modeling and predicting human behavior in games and decision-making tasks. This section reviews the most relevant works in human-agent interaction, off-policy evaluation, Multi-Armed Bandit (MAB) algorithms, and persuasion in NLP.

2.1 Off-Policy Evaluation

Off-policy evaluation (OPE) is a critical component in the design of interactive agents, enabling the assessment of an agent's performance using historical data (Precup 2000). Traditional OPE methods have been employed in recommendation systems and personalized medicine. In the context of language-based games, OPE involves predicting human decisions based on past interactions, which poses unique challenges due to the dynamic nature of language and human behavior.

2.2 Persuasion in NLP

Persuasion has been a focus in natural language processing (NLP) research for many years. It refers to the study and implementation of techniques to influence others' attitudes, beliefs, or actions through natural language. Researchers in this field develop models and algorithms to understand how language

[1] Our code can be found in the following link: https://github.com/michalbar031/NLP2024$_{project}$

can be used effectively to persuade. LLMs enhance our understanding of human behavior in non-cooperative, language-based persuasion games (Apel et al.2022). These models can predict human decisions by training on interactions between humans and artificial agents, providing insights into how humans respond to persuasive strategies. To enhance off-policy performance, a simulation technique involving interactions and simulated decision-makers was proposed.(Shapira 2024).

2.3 Human Decision-Making in MAB

The Multi-Armed Bandit (MAB) framework provides a powerful tool for making decisions under uncertainty. MAB algorithms have been widely used in online advertising, clinical trials, and adaptive learning systems.(Agrawal 2012) Human decision-making in MAB problems has been widely studied. Steyvers et al. (2009) conducted a Bayesian analysis of human decision-making in bandit problems.

3 Data

For training the model we used real data as in the original paper by Shapira et al. (2024). We modified the simulation as will be described in section 4, and then simulated more data points for training. For training the classifier and building the customized embeddings we used an external dataset "Trip Advisor Hotel Reviews" from Kaggle. [2]

4 Model

4.1 Multi-Armed Bandit Problem

The multi-armed bandit (MAB) problem is a classic problem in probability theory and decision-making, where the objective is to maximize the total reward over a sequence of trials. The challenge is to choose which arm to play at each step to maximize cumulative rewards. **The Standard Multi-Armed Bandit:** A multi-armed bandit (MAB) $\mathcal{M}$ is defined by:

- Θ: a space of reward distributions parameters; $\theta \in \Theta$;
- N: an integer representing the number of arms;
- p: a distribution over Θ.

At the start of the game, θ is sampled from Θ according to the prior p. At each timestep t, an arm $a_t \in [1, ..., N]$ is chosen. A reward $r_t \sim \theta_{a_t}$ is sampled. A strategy is a mapping that determines the correct distribution to sample from given a history of reward observations and previous arm pulls: $K_t(a_1, r_1, ..., a_{t-1}, r_{t-1})$.

μ_k represents the mean of arm k, with parameters θ_k. j^* represents the index of the best arm and μ^* to represent its mean. $T_k(t)$ represents the number of pulls of arm k up to and including time t. The goal of this game is to maximize the sum of rewards over time, or alternatively, to minimize the expectation of the regret $\overline{R}(t)$, defined as:

$$\overline{R}(t) = \sum_t (\mu^* - \mu_{a_t}) = \sum_k (\mu^* - \mu_k) T_k(t)$$

4.2 Thompson Sampling for MABs

The Thompson Sampling algorithm is a Bayesian approach to the multi-armed bandit problem that maintains a prior distribution over the mean rewards of each arm. For Bernoulli bandits, where rewards are either 0 or 1, the natural choice of prior is the Beta distribution. **Bernoulli Bandits** In the case of Bernoulli bandits, the Thompson Sampling algorithm maintains a Beta distribution with parameters α and β for each arm i, denoted as $\text{Beta}(\alpha_i, \beta_i)$. Initially, the algorithm assumes a uniform prior $\text{Beta}(1, 1)$ for all arms, which is equivalent to assuming no prior knowledge about the mean rewards. At each time step t, the algorithm samples a value $\theta_i(t)$ from the Beta distribution of each arm i. It then selects the arm $i(t)$ with the highest sampled value $\theta_i(t)$. After observing the reward $r_t \in 0, 1$, it updates the distribution parameters for the selected arm $i(t)$. If $r_t = 1$, it increments the success count $S_{i(t)}$, otherwise it increments the failure count $F_{i(t)}$. The updated Beta distribution for arm $i(t)$ becomes $\text{Beta}(S_{i(t)} + 1, F_{i(t)} + 1)$.

[2] The external dataset "Trip Advisor Hotel Reviews" can be found in: Trip Advisor Hotel Reviews

Algorithm 1 Thompson Sampling for Bernoulli Bandits

```
Data: Successes and Failures for each arm
Result: Selected Arm
for each arm $i = 1, \dots, N$ do
    Set $S_i = 0, F_i = 0$
end
for each time step $t = 1, 2, \dots$ do
    for each arm $i = 1, \dots, N$ do
        Sample $\theta_i(t)$ from Beta($S_i + 1, F_i + 1$)
    end
    Play arm $i(t) := \arg\max_i \theta_i(t)$ and observe
     reward $r_t$  if $r_t = 1$ then
        $S_{i(t)} = S_{i(t)} + 1$
    end
    else
        $F_{i(t)} = F_{i(t)} + 1$
    end
end
```

4.3 Our Simulation Setup

In our simulation, the arms corresponded to different strategies that the decision-making bots could employ. Instead of using an initial fixed probabilistic vector (that updates the probabilities each round) to choose strategies, we applied the MAB framework. Specifically, we used Thompson Sampling to dynamically select the most promising strategies based on their performance. The goal is to simulate realistic human learning processes. Furthermore, we implemented two new human-like behavior strategy functions. We created the following strategies: Classifier-based strategy, cost-benefit hybrid strategy. As well as using the Trustful Strategy, Language-Based Strategy and Random Strategy.

Cost Benefit Hybrid strategy: The function considers both the hotel's perceived quality and the user's past experiences with the bot's recommendations. If there are previous rounds, the function adjusts the perceived quality based on the outcome of the last round. Suppose the bot's last recommendation was for a hotel with a quality score of 8 or higher and the actual reviews averaged 8 or higher. In that case, the perceived quality is increased by 0.7, indicating increased trust in the bot. Otherwise, the perceived quality is decreased by 0.4, indicating decreased trust in the bot. Then it compares the adjusted perceived quality against the cost threshold. If perceived quality is greater than or equal to the cost threshold, the function returns 1. Otherwise, returns 0.

Classifier based strategy: We aimed to develop a model capable of predicting the scores of reviews based on their textual content. We employed a custom vectorization approach to generate embeddings that could capture the nuances of each review. Specifically, we utilized a combination of traditional NLP techniques such as TF-IDF, alongside more advanced methods including POS tagging distributions and sentence embeddings derived from the Universal Sentence Encoder (USE) and SBERT models. These features were integrated into a single vector representation for each review. This model leverages a transformer-based architecture, which is suited for handling natural language and capturing contextual relationships within the text. By training this model on a dataset of reviews and their corresponding scores, we aimed to enable it to predict the score of a given review. The scores are 5 classes 1,2,3,4,5. Then the strategy function uses this prediction to choose an action. **Feature Extraction for Review Embeddings:** we extract a comprehensive set of features from each review to form a robust embedding for our model. This feature extraction process is designed to capture various lexical, syntactic, semantic, sentiment, and readability characteristics of the text. Lexical Features: Word Count, Average Word, Unique Word Count Length, Character Count. Syntactic Features: Sentence Count, Average Sentence Length, Punctuation Count, Part-of-Speech (POS) Distribution. Semantic Features: TF-IDF Scores, Universal Sentence Encoder (USE) Embedding, Sentence-BERT (SBERT) Embedding, Named Entity Recognition (NER) Counts. Sentiment Features: Sentiment Polarity, Sentiment Subjectivity. And Readability Features: Flesch Reading Ease, Flesch-Kincaid Grade Level. Finally, all these features are combined into a single feature vector that serves as the embedding for the review. This embedding approach ensures that both the surface-level characteristics and deeper semantic nuances of the reviews are considered, providing a rich input for the model's decision-making process.

4.4 Implementation in the Given Context

In our study, we implemented the multi-armed bandit approach with Thompson Sampling to select strategies for simulated users in a language-based persuasion game. Our approach involves the following steps:

Initialize Parameters: Each strategy has associated success and failure counts, both initialized to zero. These counts are updated based on the outcomes observed from applying the strategy.

Select Strategy Using Thompson Sampling: For each strategy, a probability is sampled from the Beta distribution parameterized by the strategy's success and failure counts. The strategy with the highest sampled probability is selected for the current round.

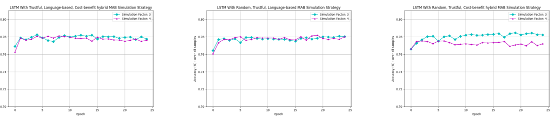
Update Counts Based on Outcomes: After applying a strategy, the outcome (success or failure) is observed. The success or failure count of the chosen strategy is updated based on this outcome.

5 Experimental Set Up

In our experiments, we utilized Weights Biases (wandb) to manage and track our model hyperparameter sweeps. The **online simulation factor**, which controls the factor for online simulation, was tested with values 4 and 3. Prioritizing certain strategies in a multi-armed bandit (MAB) setup can be particularly beneficial in scenarios where the exploration budget is limited, such as when only a small number of rounds (e.g., 10) are available. The **prioritized strategies** parameter is used to initialize certain strategies with a higher number of successes. This biases the multi-armed bandit algorithm to prefer these strategies early in the decision-making process. The predefined sets of prioritized strategies are: Set 0: Trustful, Language-based, Cost-benefit hybrid Set 1: Random, Trustful, Language-based Set 2: Random, Trustful, Cost-benefit hybrid.

6 Experiments and Results

In Figure 1, we evaluated the performance of various simulation strategies using LSTM architecture, comparing our results to those presented in the original paper. MAB Simulation Strategy 0, uses: Random, Trustful, Language-based. The accuracy ranged from 0.76 to 0.78 across epochs, stabilizing after initial fluctuations. The trends for both simulation factors (3 and 4) were close, indicating robust performance. Compared to the original paper, this strategy shows lower absolute accuracy but demonstrates consistent stability. Strategy 1 incorporates cost-benefit analysis alongside random and trustful MAB simulations, aiming to check our learning strategy "cost-benefit" compared to the original Language-based strategy. Both simulation factors showed similar trends, with accuracy stabilizing around 0.78, showing marginal improvement with factor 3. While the results did not surpass those of the original paper, they indicate that cost-benefit analysis can be beneficial, particularly with the right simulation factor and showed slightly better results compared to the Strategy 0 setup. Strategy 2 combines trustful behavior with language-based decision-making and cost-benefit. The simulation that used factor 3 showed a slight upward trend, consistently outperforming factor 4. To sum up, he original paper's strategies achieved higher accuracy levels, ranging from 0.81 to 0.84, compared to our results which ranged from 0.76 to 0.78. However, our strategies might exhibit greater stability.



(a) Strategy 0 (b) Strategy 1 (c) Strategy 2

Figure 1: We evaluated the performance of various simulation strategies using LSTM architecture. We show the accuracy achieved by each of the examined simulations.

7 Conclution

Our experiments demonstrate that while the current MAB setup and strategies do not surpass the accuracy levels reported in the original paper, they offer stable performance across epochs. Future work should focus on optimizing simulation strategies and exploring new combinations to bridge the accuracy gap while maintaining stability. In conclusion, while our results show lower absolute accuracy, the possible increased stability is notable.

References

Apel, R., Erev, I., Reichart, R., Tennenholtz, M. (2022). Predicting decisions in language based persuasion games. Journal of Artificial Intelligence Research, 73, 1025-1091.

Shapira, E., Apel, R., Tennenholtz, M., Reichart, R. (2024). Human Choice Prediction in Non-Cooperative Games: Simulation-based Off-Policy Evaluation. arXiv preprint arXiv:2305.10361.

Precup, D. (2000). Eligibility traces for off-policy policy evaluation. Computer Science Department Faculty Publication Series, 80.

Barron, G., Erev, I. (2003). Small feedback-based decisions and their limited correspondence to description-based decisions. Journal of behavioral decision making, 16(3), 215-233.

Agrawal, S., Goyal, N. (2012, June). Analysis of thompson sampling for the multi-armed bandit problem. In Conference on learning theory (pp. 39-1). JMLR Workshop and Conference Proceedings.

Steyvers, M., Lee, M. D., Wagenmakers, E. J. (2009). A Bayesian analysis of human decision-making on bandit problems. Journal of mathematical psychology, 53(3), 168-179.