Surviving Tinder: A Decision Theoretic Modeling of Tinder via POMDPs

Jihee Hwang
Stanford University

Yong Nam Kwon
Stanford University

Ho Kyung Sung
Stanford University

Abstract

The world of Tinder is ever intriguing yet mysterious. We present an efficient abstraction of this widely adopted dating app , and a problem-solving agent optimizing for best match results. We model the problem as a partially observable MDP (POMDP) in which the agent makes one of three choices: like, dislike, or change profile. By incorporating various considerations about the latent and observable variables that influence the observable process into the game model, we show that a utility-maximizing agent performs well on getting the most out of Tinder, in comparison to random and greedy policies.

1. Introduction

In the world of dating apps, numerous variables dictate the probability of successfully matching with a suitable partner. Conflicting constraints such as the perceived attractiveness of the opposing party and the uncertainty as to whether the romantic interest is reciprocated, all work together to confound the decision making process, and oftentimes force one to make sub-optimal choices. Having been unsuccessful players at the intriguing game Tinder offers, our goal in this paper is to propose a decision theoretic model framework that incorporates the various constraints and variables to guide the user's actions in a dating app. The hope is that our model accurately simulates the decision making environment of dating apps and can possibly suggest optimal or near-optimal policies to harvest maximum utility out of the app.

Previous research have presented solutions to a similar albeit distinct problem, namely the optimal stopping problem [2][3], in which the goal is to optimize the point at which one stops exploring until one finds a decent candidate and decides to settle. However, at least under its classical formulation, the optimal stopping problem is unsuitable for modeling the dynamics of dating apps in two critical respects: first, the optimal stopping problem assumes that we only make one decision for all possible candidates; second, the problem assumes that our decision will always be met with approval on the part of the opposing party.

As such, we must further complicate the problem and attempt to design a model that relaxes these assumptions, allowing a richer, more flexible, and therefore more realistic representation of the dating app environment.

2. Background

2.1. Tinder

Tinder is one of the most widely used dating apps around the world. Every day, users are allowed to search through potential matches and express their liking for a limited number of times. By swiping “right” on a profile, the user can “like” the opposing party, while by swiping “left”, the user moves on to the next candidate.[1] Though Tinder has gained its popularity through its straightforward and easy-to-use user interface, it is not as easy to be successful. According to statistics released in 2014, there are more than 1 billion swipes performed daily, while only 12 million matches are made.[5]

There are many variables, both latent and observable, that affect the decision making process on the Tinder app. A few among those are: the attractiveness of the user, the attractiveness of the potential match, the strength of the appeal of the user's profile, the remaining “swipes” of the day and the remaining chances to ”like” the potential match. Our model of Tinder, if it is to be realistic, must incorporate and account for these variables as faithfully as possible.

2.2. Tinder as a Multi-Armed Bandit Problem

One of the simplest models that can be applied to the Tinder environment is that of a multi-armed bandit, an MDP with a constant state.[6] Assuming that the user has a choice between n profiles of one's own, and assuming that each of the profiles have a constant probability of being “liked” by the opponent, the problem of choosing which profile to display is an almost exact application of a multi-armed bandit problem. The solution to the multi-armed bandit requires for a delicate balance between exploration and exploitation. In other words, the user must choose to try out several profiles in order to discover the profile that garners the most likes, then finally exploiting the information from the exploration.

However, the dating landscape is fundamentally a two-way interaction, with plenty of unpredictable factors. It would be very naive to assume of course that a given profile, no matter attractive, will always yield a constant amount of matches. Therefore we can further relax this assumption by modeling the problem as a POMDP.

2.3. Tinder as a POMDP

As can be easily noted, there are critical variables that the multi-armed bandit setting fails to account for. For one thing, an appropriate model must be able to make a suggestion as to whether to"swipe right" or "swipe left" the current candidate. In addition, the model should take into account the perceived attractiveness of the opposing party as an important factor in the decision making process, rather than unilaterally perceiving all matches as having the same utility. Finally, the model must be able to account for the user profile's actual attractiveness to the candidates, as well as the user's belief about their own attractiveness.

As such, we propose that a partially observable MDP (POMDP) is best fit to model decision making in Tinder. POMDPs represent unobservable variables via hidden states, which are partially estimated by observations and the beliefs inferred from the observations. Also, there is ample flexibility in which we can incorporate various constraints and factors into the problem definition[4]. In the following chapters, we present a specific definition and implementation for a Tinder-POMDP, and results of experiments on the model.

3. Methods

3.1. Problem Model

We define the following variables.

- $c_{\text{user}}, c_{\text{candidate}}$: The degree of attractiveness of the user and the candidate, respectively.
  $c_{\text{user}}, c_{\text{candidate}} \in \{1, 2, 3, \cdots 10\}$, where 1 represents minimal attractiveness and 10 represents significant attractiveness.
- h: Length of the total horizon (Number of available swipes given daily).
- h_t: Remaining number of swipes at a certain timestep.
- l: The total number of "likes" or right swipes the user can expend daily.
- l_t The remaining number of "likes" or right swipes at a certain time step.

3.1.1 Key Assumptions

The goal of the POMDP agent is to maximize the value of matches given to attractive counterparts, given a finite number of swipes. Every user's profile implies a certain degree of attractiveness that is assumed to be objectively assesable as a natural value. Just like how one would in the real app Tinder, the agent needs to take a series of strategic actions in order to maximize the number of good matches within the given number of tries.

3.1.2 Actions and State Space

We have modeled player's actions as being one of three possible choices: like or dislike a candidate, or change one's profile. By trying out different profile pictures and self-introductions, the user can choose to explore different states with different attractiveness. If the user changes one's profile, the user's attractiveness can move in any direction within a range of 2. The user can only take h actions and send l likes.

Therefore, the state consists of 4 possible variables: the player's attractiveness, the counter party's attractiveness, remaining horizon (tries), and remaining number of likes.

3.1.3 Observation Probability

The probability p of getting a match after "liking" a candidate has a positive correlation with c_{user} and a negative correlation with $c_{\text{candidate}}$. The reward is proportional to $c^2_{\text{candidate}}$ and inversely proportional to c_{user}, which was modeled in effort to take into consideration that people may perceive a greater amount of reward once they are matched with someone more attractive than they are. Logarithmic and linear transformations are applied to ensure that the returning probability is a value between 0 and 1.

3.1.4 Reward Function

As a rational user would act in a way that would maximize utility, we introduce a cost function that takes into account the remaining number of likes and the remaining horizon. To do so, we additionally include a model for "urgentness" as well as "carefulness" in the cost function. The exact formulation we have used for the experiments are further described in detail below.

- States: A length 4 vector that has entries $\{c_{\text{user}}, c_{\text{candidate}}, h_t, l_t\}$ $c_{\text{user}}, c_{\text{candidate}} \in \{1, 2, 3, ..10\}$
- Actions: $a \in \{\textit{Like}, \textit{Dislike}, \textit{Change profile}\}$
- $\text{Reward}_{\text{like}} = \dfrac{c_{\text{candidate}}}{\log_{10} c_{\text{user}} + 1} - \dfrac{l}{h} \cdot \dfrac{h_t - l_t}{l_t}$
- $\text{Reward}_{\text{dislike}} = 0$
- $\text{Reward}_{\text{change}} = \rho \approx 0$ (change inducing factor, small)

- Probability of getting matched after giving a "ike":

$$P = \frac{1}{3}(\log_{10}(\frac{c_{\text{user}}^2}{c_{\text{candidate}}}) + 1)$$

- Transition probabilities where C denotes reachable range via $change\_profile\_picture$:

$$\text{T}(\text{c}_{candidate}) = X \sim \mathcal{N}(\mu, \sigma^2)\,. \quad \text{where } \mu = 5.5, \sigma = 2$$

$$\text{T}(\text{c}_{\text{user}}) = \begin{cases} \frac{1}{\text{len}(C)} & \text{if } a = change \text{ and } new\_attr \in C \\ 0 & \text{otherwise} \end{cases}$$

$$\text{T}(\text{h}_t) = \begin{cases} 1 & \text{if } h_t - 1 \\ 0 & \text{otherwise} \end{cases}$$

$$\text{T}(\text{l}_t) = \begin{cases} 1 & \text{if } a = like \text{ and } l_t - 1 \\ 0 & \text{otherwise} \end{cases}$$

As the cost/rewards for "likes" and "change profile" actions are designed to incentivize (or decentivize) the respective actions, we ignore it for the purpose of scoring our final result. In other words, our final computed score for a simulated run of the MDP is a sum of the match-rewards over all the matches made in that simulation.

3.2. Policy Search Methods

We choose online methods for determining policies over other methods, largely for the following reasons: first, the state space is too large for an exact or offline solution to be accurately derived; second, the state variable includes random samples from a normal distribution (for the candidate's attractiveness) that is stochastic and can be specified only at each time step, rather than offline; finally, actual Tinder users would adjust their policy through online methods as well. We implement the forward search algorithm, and tested with forward search of depth 1 (one-step lookahead) and 2.

The equation[4] for computing the expected utility for action a is given as:

$$R(b, a) + \gamma \sum_{0} P(o \mid b, a) U_{d-1}(\text{UPDATEBELIEF}(b, a, o))$$

where $U_{d-1}(b')$ is the expected utility returned by the recursive call to find the maximum expected utility action from the next belief state. We choose the action that has the maximum expected utility.

4. Experimentation

All experiments were conducted through Python3.
The code can be found in
https://github.com/ynkwon/TinderPOMDP.

4.1. Data

Due to the sensitive nature of the involved data, it was impossible to obtain real-life data directly from Tinder. As such, we generated our own pseudo-data for the purposes of testing our model. Based on the various observation (match) and transition probabilities that we have designed, we generate a series of tinder interactions under varied assumptions regarding the condition. One of our assumptions is that the attractiveness of a tinder candidate would follow a normal distribution, just as height tends to be distributed across a given population, with mean of 5.5 and standard deviation of 2.

4.2. Experiment Procedure

We tested our models and several policies against mainly two different variables: the player's true attractiveness (regardless of its beliefs) and the player's initial belief about one's attractiveness.

4.2.1 Comparison of Various Policies

In order to test the robustness of our model, as well as our online policy searching method, we compared the result of simulation runs on the same dataset over the following policies: random, greedy, 1-step lookahead, and 2-step lookahead. Our random policy randomly chose one of the three actions with equal probability. Our greedy policy chose to blindly like the candidate if the candidate's attraction was greater or equal to 6; otherwise our choice of action was to dislike. Our 1-step and 2-step lookahead policies followed the implementation of online methods outlined in section 3.2.[4]

4.2.2 Effects of Different Initial Beliefs

As our belief state is an estimation of our actual attractiveness, we wanted to test what effects the initial belief state (and its discrepancy to the true hidden state) had to the overall result. Ideally, after sufficient iterations, the belief state should converge or at least be consistent with the actual hidden state, which means that the long-term expected utility must be similar. However, under the current parameters, this might not be the case, and we wanted to examine if there was a noticeable pattern depending on the initial belief.

For the experiment, we tested with three different kinds of initial beliefs. The uniform initial belief implies that the user is completely unaware of where their

Starting True Attraction	3					
Initial Belief	**Uniform**		**High-Skewed**		**Low-Skewed**	
Results	**Total Reward**	**Matches**	**Total Reward**	**Matches**	**Total Reward**	**Matches**
Random	16.92	6	20.31	7	27.08	8
Greedy	28.43	6	23.69	5	13.54	3
One-step Forward Search	38.59	9	38.59	9	27.76	7
Two-step Forward Search	38.59	9	39.94	9	35.88	8

Starting True Attraction	7					
Initial Belief	**Uniform**		**High-Skewed**		**Low-Skewed**	
Results	**Total Reward**	**Matches**	**Total Reward**	**Matches**	**Total Reward**	**Matches**
Random	14.63	7	5.96	2	35.23	14
Greedy	46.06	13	44.98	12	39.02	11
One-step Forward Search	42.82	12	44.44	13	37.40	11
Two-step Forward Search	37.94	11	44.98	12	37.94	11

Table 1: Experiment results.

attractiveness stands, and thus the prior is given as $[0.1, 0.1, \cdots, 0.1]$. The high-skewed initial belief implies the user think of themselves as being highly attractive. For the experiment, the initial belief distribution of [0.01,0.01,0.01,0.01,0.01,0.01,0.01,0.09,0.8,0.1] has been used. Last, the low-skewed initial belief implies the user think of themselves to be unattractive. The initial belief of [0.1,0.8,0.09,0.01,0.01,0.01,0.01,0.01,0.01,0.01] has been used for the low-skewed experiment. As the random and greedy policies make no use of the belief state in any way, the only relevant policies for this part of experimentation were the 1-step and 2-step lookahead policies.

5. Results

Our results from the experiments are shown in Table 1 above. The models had parameters of $h = 100$ and $l = 20$, and have been experimented across different starting attractions, 3 different initial belief states, and 4 different policies. Additionally, we have examined the progress of the agent's belief states as it advances along each trial.

From the figures presented, we can confirm that the belief update function of the POMDP is working exactly as expected. Upon observing the varied results after giving a like, the user successfully alters their assumed beliefs on their own attractiveness. Figure 1 shows that after being accepted by the counter party, the user shifts one's beliefs to the left; in other words, they are inclined to think that their attraction is actually lower than what they expected. Notice that after the correction, the probability distribution tends to get more spread out. This happens likely because the lower the attraction, the more likely one is going to be unmatched with the candidate, therefore leading to a slightly higher probability increase.

Figure 2 shows the opposite situation in which the user

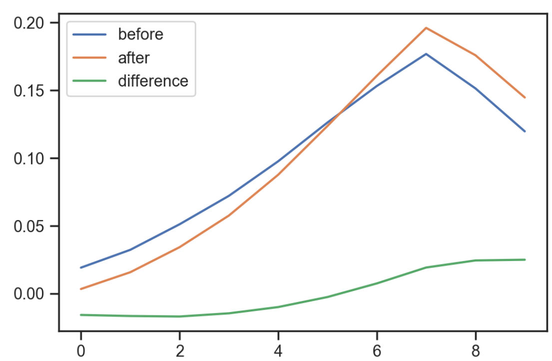

Figure 1: Belief update: right shift effect after an ”matched” result.

is successfully matched after giving a like. The user is likely more confident (albeit slightly so) and therefore adjusts one's expectations in such way.

The experiment of figure 3 was conducted with an initial true attractiveness of 7. Because of the stochastic nature of the matching process, the belief has been updated to spread the probability across different states. However, the simulation accurately ended with a finishing belief that has its peak value at an attraction level of 7, which is the ground truth value.

6. Discussion and Analysis

6.1. Explanation and Interpretation of Results

In general, there was a consistent pattern in which the expected-utility-maximizing agents performed better or at least as well compared to agents abiding by random or

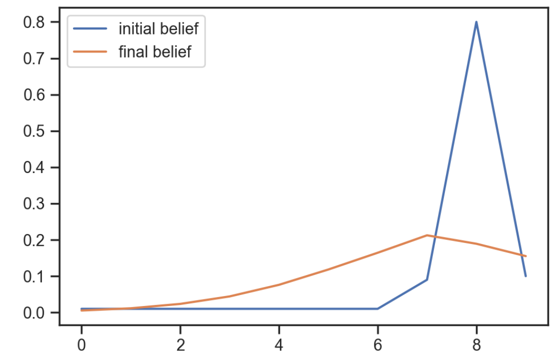

Figure 2: Belief update: slight converge effect after 100 iterations

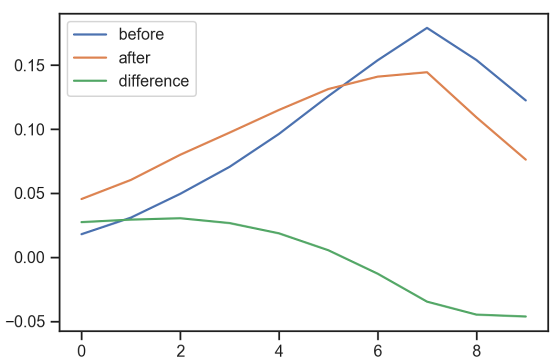

Figure 3: Belief update: left shift effect after an "unmatched" result.

greedy policies. This signifies that the reward and transition probabilities of the POMDP were relatively adequately defined to enable near-optimal decision that increased overall utility. As the figures of the belief update show, the model seem to have correctly represented the relevant relationship between latent and observed variables.

In particular, an interesting pattern was found with the greedy policy. The greedy-policy generated actions yielded relatively high rewards, especially when the true attraction was high (true attraction = 7). This is most likely explained by the fact that for an agent with high true attraction and similar self-confidence, the actions suggested by utilizing-maximizing agents is often the same as the actions that are chosen by the greedy policy.

Another interesting pattern that we observed was that independent from one's true attractiveness, the average reward of policies was higher when the user had a high-skewed initial belief. This pattern was even more apparent when the forward search policy was used. This suggests that having a higher level of confidence makes the user to be more eager to like candidates of higher attraction scores, which brings in general better results compared to having a low level of confidence.

Averaging the total reward and number of matches with respect to the user's true attraction, a true attraction of 3 gave about a mean total reward of 29.11 and an average matching of 7.16, while a true attraction of 7 had values of 35.95 and 10.75, respectively. Thus, we can reaffirm that a higher true attraction indeed returns better reward and more matches. What follows perhaps is a humbling truism, namely that those who are more attractive are indeed more likely to succeed in dating app environments.

6.2. Limitations and Future directions

The algorithms of Tinder are inherently secretive due to its sensitive nature. None of Tinder's source code or API is revealed, therefore close to none can be inferred of its inner workings. Naturally, we had to adapt numerous assumptions when modeling the nature of Tinder.

The model we have come up with has an underlying assumption that attractiveness is objective and absolute. However, different people have different preferences. To accommodate this fact, we could choose to instead adapt a normally distributed probability distribution of what one may evaluate the candidate's attractiveness.

Also, we have modeled the reward function under the assumption that people receive a higher reward once matched with a candidate with a higher attractiveness score. However, it is also intuitively agreeable that people may be as happy as, if not happier, once matched with a candidate with a similar attractiveness score compared to being matched with a candidate who has a higher attractiveness score. With a comprehensive survey addressing this question, we would be able to create a model that more closely reflects the perceived utility of Tinder users.

Last, although we have modeled assuming that all candidates immediately respond with either a rejection or an approval, matches in Tinder are made only when both the user and the candidate have seen and liked each other. In order to account for whether the opposing party has seen one's own profile and decided on it, we would have to incorporate consideration of another hidden state or some probabilistic factor. Adding such a feature to our model would allow us to design a decision algorithm that acts robustly even under minimal immediate feedback/response, as is the case in actual dating app environments.

In terms of actual implementation to Tinder environments, the last important factor that is wholly missing in the current implementation is a sensible estimator of the attractiveness of real people's profiles.

7. Conclusion

We have presented a POMDP model of decision making in Tinder. As a result of simulating and experimenting on the POMDP under various circumstances, we poignantly conclude the following: First, we observed that generally the expected-utility-maximizing policies indeed generated the best results. Second, unsurprisingly, greedy policies generate high rewards given the user has a high true attractiveness. Third, having high confidence (high initial beliefs) in one's attractiveness tend to generate better results. Fourth, having a high true attractiveness returns better rewards and enables more matches.

8. Contribution of Group Members

All members of the group contributed equally. Yong Nam took charge of POMDP modeling. Jihee played a major role in implementing the defined POMDP in Python. Ho Kyung simulated the MDP, carried out the the experiments, and derived the results.

References

[1] Tinder (app), Dec 2019.

[2] B. Christian and T. Griffiths. *Algorithms to live by: the computer science of human decisions*. Henry Holt and Company, 2016.

[3] Y. Girdhar and G. Dudek. Optimal online data sampling or how to hire the best secretaries. In *2009 Canadian Conference on Computer and Robot Vision*, pages 292–298. IEEE, 2009.

[4] M. J. Kochenderfer. *Decision making under uncertainty: theory and application*. The MIT Press, 2015.

[5] A. Shontell. Nope: People are getting rejected hundreds of millions of times on tinder every day, Oct 2014.

[6] J. Vermorel and M. Mohri. Multi-armed bandit algorithms and empirical evaluation. In *European conference on machine learning*, pages 437–448. Springer, 2005.