Rizz Is All You Need: Expert Dating via Reinforcement Learning

Fan Pu Zeng [1]

Abstract

We solve the age-old problem of dating by applying algorithms and techniques from the reinforcement learning community. We perform a theoretical analysis on how this allows us to converge to becoming a person with rizz, and conclude with a call-to-action for the community to help to empirically verify these theoretical guarantees by trying them out and sharing their experiences.

1. Introduction

There has been a flurry of exciting advancements in reinforcement learning in the last few years, ranging from using deep reinforcement learning to achieve super-human performance on Atari in 2013 (Mnih et al., 2013), to the defeat of the 9 dan Go world champion Lee Sedol by AlphaGo in 2016 (Silver et al., 2016), to the development of Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022) techniques that has resulted in the success of large language models like GPT-4 in 2023 (OpenAI, 2023). It is increasingly the case that AI can achieve super-human performance on many tasks traditionally thought to be the forte of humans.

In spite of these advances, there has not yet been a systematic study of how such reinforcement learning techniques can also be applied to our personal lives. One of the hardest tasks faced by computer scientists today is to leave their computers and to attempt to engage in face-to-face communication. The stakes become exponentially higher once the goal of the interaction is to pursue a romantic outcome with the counterparty, during which many people fall into the trap of analysis paralysis while optimizing for their respective objective functions.

In this paper, we tackle this age-old dating problem head-on once and for all by presenting how novel applications of reinforcement learning techniques can be used to solve it.

[1]School of Computer Science, Carnegie Mellon University, Pittsburgh, USA. Correspondence to: Fan Pu Zeng <fzeng@andrew.cmu.edu>.

Proceedings of SIGBOVIK [0x2023], Pittsburgh, Pennsylvania, USA 15213,

2. Background

We can model our problem setup as the tuple $(\mathcal{S}, \mathcal{A}, p, r, \gamma)$, where:

- $\mathcal{S}$ is the set of all possible states, which is how the other party currently feels about you.
- $\mathcal{A}$ is the set of all (legal) actions that you can take.
- $p(s' \mid s, a)$ is the unknown Markov Decision Process (MDP) that returns the probability that your crush will transition to state s' starting from state s after you take action a.
- $r(s, a) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function that determines the reward that you get from taking action a when the other person has feelings at state s about you. Note that rewards can be negative (i.e if you oversleep your date because you were grinding a PSET due last night and the other person gets mad).
- $\gamma \in [0, 1]$ is the discount factor based on how much you value future rewards vs immediate rewards as we model our setup as an infinite horizon problem. In practice, since we are only on this earth for a limited amount of time and we don't know when we are going to die (otherwise it becomes a finite-horizon problem), this should be less than 1.

Our goal is to learn a policy π that maximizes our expected reward, thereby winning the other person's heart. For any state s, $\pi(s)$ is a distribution over actions to take. So at any time step t, the probability that our next action A_t given current state S_t are a and s respectively based on our policy is

$$\pi(a \mid s) = \mathbf{Pr}\left(A_t = a \mid S_t = s\right). \tag{1}$$

Furthermore, we introduce the notion of state-value function $Q_\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ and action-value function $V_\pi : \mathcal{S} \to \mathbb{R}$, both paramterized by our policy π. Letting G_t be the total future discounted reward at time step t, we can then formally

define them

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}, \tag{2}$$

$$V_\pi(s) = \mathbf{E}\left[G_t \mid S_t = s\right], \tag{3}$$

$$Q_\pi(s,a) = \mathbf{E}\left[G_t \mid S_t = s, A_t = a\right]. \tag{4}$$

3. Related Work

There is a dearth of suitable resources available on how to solve the dating problem. Classical approaches include asking (somehow usually single) friends for advice, reading self-help books, remembering pick-up lines from TV shows, Googling, etc. More modern approaches include asking your followers on your Finsta and TikTok for advice, taking your crush out to SCS day, cold-emailing your academic advisor for good date ideas, etc. Unfortunately, empirically all of these approaches have been met with limited success, and none of them come with any theoretical guarantees.

4. Approaches

4.1. ε-Greedy

When talking to someone that you like, it is often difficult to say anything other than the safest and most reliable action in case you end up humiliating yourself or offend the other person. This effectively means always choosing your action with the best Q-estimate, such as asking your crush if they would like to work on the 15-451 HW together after class today.

But as the literature points out, without sometimes trying other strategies, you will never be able to update your Q-estimates and explore other potential promising strategies (the exploration vs exploitation tradeoff), and therefore it makes sense to randomize your actions ε fraction of the time. This is also important in the presence of distributional shift, since our estimates of other states may have become stale. So next time consider trying something completely different and see how that works out!

4.2. Updating Value Function Estimates

How do you know whether your first date went well? Or the real question is, how do you tell if you said the right things and made jokes that your crush actually found funny and they weren't just laughing out of courtesy?

The literature suggests two primary ways of doing this. The first approach is known as Monte-Carlo sampling, where you collect an entire trajectory of experiences (i.e perhaps until you get married to the person, or otherwise never talk to the person again because they hate you so much), and then update your V estimates accordingly (either positively or negatively depending on the outcome). The downside to this approach is that it's probably going to take you years before you figure out whether you did the right thing, and it means we won't get a whole lot of chances to learn our value function before we die.

Let's try something else. The n-step temporal difference (TD) approach allows us to update our value estimates after taking just n extra steps from state s_t to s_{t+n}, by making use of our bootstrapped V estimate $V_\pi(s_{t+n})$ to estimate the future rewards of the final state. This allows us to incrementally update our estimates, which converges faster and lets us make use of our new knowledge earlier. In other words, if your date texts you that they had a good time and that you both should hang out again soon, you're probably doing something right.

4.3. Proximal Policy Optimization

You took our advice in 4.1 to heart, and mustered the courage to do something you would have never imagined you would ever do: you gave your crush a cheeky slap across the face as you met them at Fuku Tea for your second date. And guess what, they are now mad, and angrily stormed off while mentioning something about Title IX. It appears that you have put yourself in an irrecoverable state where nothing can save you now.

What did we do wrong here? As it turns out, we tried to do something too wild, and now we are paying the price for it! We can avoid this by using proximal policy optimization (PPO) algorithms (Schulman et al., 2017) when updating our policy. In PPO methods, whenever we update our policy by updating it based on the gradient of the current policy multiplied by a stepsize (like gradient descent), we also add an additional regularization term that takes into account the Kullback–Leibler (KL) divergence between the two policies, and therefore penalizes new policies that are too different from the current policy. This ensures that while we do still try new things to woo the love of our life, we are not going to do something so drastically different from our current approach that could lead to disastrous consequences.

4.4. Imitation Learning

You decide to engage in imitation learning and start watching romantic films to learn from the best. Except that your best friend points out, what if the other person responds differently from what happens in the movie? What do you do next? There are no reference solutions to take inspiration from! Now you're in deep trouble since you're in a state that is completely off the path of expert demonstration trajectories.

4.5. Dataset Aggregation (DAGGER)

"Let me help you!", your roommate says. They agree to pretend to be your date, and train your rizz by performing simulations based on the movie scenarios together, and of course introducing more variability in what actually pans out. After each run, they (as highly qualified dating advice experts) point out your mistakes and show you how to correct them. This is like the Data Aggregation (DAGGER) method (Ross et al., 2011), where you have an expert to help correct mistakes that you might plausibly encounter during your own trajectories, that previously did not exist in any of the expert demonstration training trajectories.

4.6. Sim2Real Transfer

By some stroke of luck smaller than the chances of an asteroid wiping out the planet, you managed to ask your crush out on a date again, and it actually went pretty well this time! In fact, things are going so well that you realized that soon you may even get your first kiss!

This terrifies you beyond words, because you have no idea how to do it and don't want to mess it up. However, you suddenly recalled that some very brilliant researchers here at CMU recently developed a VR haptic system that allows you to simulate, yes you guessed it, a kiss! (Shen et al., 2022) You try on the contraption, and train your kissing abilities over all possible configurations of the mouth-bound haptic device, even if they are completely unrealistic! But this is indeed the approach adopted by sim2real transfer (OpenAI et al., 2019), which prepares you for any possible scenario, building both your abilities and confidence in yourself.

5. Experiments

We are currently actively looking for volunteers to take part in this study in order to collect experimental results. Volunteers will be rewarded generously for their time and bravery.

6. Future Work

One limitation of being just a single person is that we can only perform training sequentially. It would be interesting to consider how large-scale parallel training can be performed if we can share our collective consciousness and experiences.

7. Conclusion

We show how modern reinforcement learning techniques are not just limited to virtual agents on a computer, but can also be applied to ourselves in real life, and show how theoretically we can train ourselves to become an expert in dating with good sample efficiency, perhaps even in this lifetime.

The author absolves himself from any and all liabilities resulting from any damages or loss incurred due to following any of the recommendations in this paper.

References

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing atari with deep reinforcement learning, 2013.

OpenAI. Gpt-4 technical report, 2023.

OpenAI, Akkaya, I., Andrychowicz, M., Chociej, M., Litwin, M., McGrew, B., Petron, A., Paino, A., Plappert, M., Powell, G., Ribas, R., Schneider, J., Tezak, N., Tworek, J., Welinder, P., Weng, L., Yuan, Q., Zaremba, W., and Zhang, L. Solving rubik's cube with a robot hand, 2019.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., and Lowe, R. Training language models to follow instructions with human feedback, 2022.

Ross, S., Gordon, G. J., and Bagnell, J. A. A reduction of imitation learning and structured prediction to no-regret online learning, 2011.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms, 2017.

Shen, V., Shultz, C., and Harrison, C. Mouth haptics in vr using a headset ultrasound phased array. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450391573. doi: 10.1145/3491102.3501960. URL https://doi.org/10.1145/3491102.3501960.

Silver, D., Huang, A., Maddison, C., Guez, A., Sifre, L., Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–489, 01 2016. doi: 10.1038/nature16961.