Timing Rating Requests for Maximizing Obtained Rating

Extended Abstract

Guy Cohen
Bar-Ilan University
cohenguy1@gmail.com

David Sarne
Bar-Ilan University
sarned@cs.biu.ac.il

KEYWORDS

Human-Computer Interaction

ACM Reference Format:
Guy Cohen and David Sarne. 2018. Timing Rating Requests for Maximizing Obtained Rating. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018,* IFAAMAS, 3 pages.

1 INTRODUCTION

The number of apps available for download in leading app stores (Google, Apple, Windows and Amazon) has reached seven million in 2017 (according to statista.com), allowing users an ever rich selection in almost any possible category. The overwhelming number of alternatives available for each app complicates the selection process, as differences are hard to spot based on the long texts specifying the apps' description and included functionalities. Here, many find the app rating systems, that have become an integral part of almost any major distribution platform nowadays, to be of much help. These rating systems allow users to provide feedback on apps they have already downloaded and used, through posting a numeric rating (typically in the scale of 1-5 or 1-10) along with a textual review message. Indeed, user rating was found to be highly correlated with app downloads and is considered by many to be the most significant indicator to an app's success [12].

Although users can rate an app whenever they want, it is often the app itself that urges them to submit a rating and user rating is typically limited to issuing a single rating for each app. Timing requests for rating is thus an important problem, as requesting at the wrong time may result in poor rating.

Naturally a user's rating is tightly correlated with how good was her experience with the app and her general satisfaction from its functionalities. In the following paragraphs we propose and report the results of an experimental evaluation of a new method for timing a request for user rating in systems that are used repeatedly, whenever the system can reason about the quality of the experience the user had within each use. The goal is to time the requests in a way that the average rating obtained is maximized over all users. The method relies on two primary principles. The first is estimating the user's satisfaction at any point, based on the quality of all prior interactions she had with the system (with an exponentially diminishing weight). The second is the modeling of the problem as an optimal stopping problem and extracting the optimal rule for initiating a rating request, such that the expected user satisfaction at the time of issuing the request is maximized.

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden. © 2018 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

Despite the importance of timing requests for rating, to the best of our knowledge the question of dynamically determining timing has not been addressed scientifically to date. Some work studying the effect of different static rating requests timings (e.g., at the beginning or end of semester, in the context of students' ratings) is available, though no significant effect of timing is reported there [5, 8, 9]. In contrast to this literature, the analysis of the results obtained in our experiments, using a dynamic timing method, reveals much influence of the timing of rating requests over the rating obtained. Timing suggestions as well as much related discussions in the context of app rating can be found in informal venues such as Internet forums and Blogs [4, 10, 13, 16]. The volume of related discussions there implies that this is a common problem developers are struggling with. We note that many recent works have suggested designs for collaborative interfaces that make use of the results of their prior actions when reasoning about the advice they should provide the user with next, in settings where the interaction with the user is inherently repeated [1, 2, 17]. In particular, it has been studied how the provision of a sub-optimal advice can improve the user's satisfaction from the agent [3, 7, 17]. While these works aim to implicitly incorporate the effect of prior actions into user satisfaction estimation, none of them relates to the decision of issuing a rating request and its timing.

2 MODEL

The model considers an agent that offers a well defined service or functionality to its users. The agent is used repeatedly, providing service N times. Each time it is being used (denoted "interaction" onwards), the agent can accurately estimate the quality of the interaction with the user (i.e., the user experience), using a well defined measure v (e.g., profit, time spent playing the game). The model assumes that the quality of any given interaction is independent of the quality of former interactions. In particular, it is assumed that the value of v is a priori uncertain and associated with a probability function $p(v)$. The agent is assumed to be acquainted with the underlying probability function $p(v)$ and, as mentioned above, gets an accurate reading of the quality of any prior interaction.

The model assumes that the agent can issue a rating request at the end of any of the N interactions with the user, requesting her to provide a numerical rating which captures her satisfaction with the agent providing the service. Furthermore, for simplicity, we assume that upon request the user will provide rating and limit the requester to requesting exactly once throughout the process.

3 THE EXPONENTIAL-SMOOTHING BASED METHOD (ESB)

With ESB, the underlying assumption is that the user's rating is influenced by the results of all prior interactions rather than merely the last one. Prior literature from psychology provides much evidence

for the claim that the effect of past experiences over one's set of beliefs diminishes exponentially as time goes by [6, 14]. Therefore a natural modeling technique for the user's satisfaction with the agent at any specific time is exponential smoothing [11, 15]. Formally, the user's satisfaction after observing the performance of the agent in the i-th interaction, denoted τ_i, is given by $\tau_i = \alpha v_i + (1-\alpha) \cdot \tau_{i-1}$, where α is the exponential smoothing parameter ($0 \leq \alpha \leq 1$), v_i is the quality of the i-th interaction and $\tau_1 = v_1$. In order to determine the proper value for α, we use the Pearson Correlation coefficient when comparing the value of τ (based on different α values) with the corresponding ratings received from users.

The goal is thus to maximize the expected value of τ at the time of the rating request. This is an optimal-stopping problem and obviously the optimal solution is threshold-based. The problem of finding the optimal set of thresholds $T_1, ..., T_N$ is complicated by the fact that the user is influenced by the entire set of observations captured by τ rather than exclusively by the last observation.

If the current interaction is the last interaction then obviously a rating request should be issued, hence $T_N = -\infty$. For any $i < N$, the agent should request rating only if $\tau_i \geq E^{\tau}_{i+1}(\tau_i)$ where $E^{\tau}_{i+1}(\tau)$ is the expected user satisfaction at the time of request if avoiding a rating request at the i-th interaction and deciding on the future rating request optimally. Otherwise, the agent should delay the rating request to one of the subsequent interactions. The value of $E^{\tau}_i(\tau)$ can be calculated using: $E^{\tau}_i(\tau) = \sum_{v|(\alpha v+(1-\alpha)\tau)\geq T_i} p(v) \cdot (\alpha v+(1-\alpha)\tau) + \sum_{v|(\alpha v+(1-\alpha)\tau)<T_i} p(v) \cdot E^{\tau}_{i+1}(\alpha v+(1-\alpha)\tau)$, where $E^{\tau}_N(\tau) = \sum_v p(v) \cdot (\alpha v + (1-\alpha)\tau)$. Meaning that both T_i and $E^{\tau}_i(\tau)$ can be calculated based on $T_{i+1}, ..., T_{N-1}$ and $E^{\tau}_{i+1}(\tau), ..., E^{\tau}_N(\tau)$ and theoretically a standard backward induction process can be applied. This however does require extracting the value of $E^{\tau}_i(\tau)$ for every possible τ value. Alas, the potential number of values τ may obtain at the i-th interaction is exponential in the number of quality outcomes. Formally, if there are k possible quality outcomes, the theoretical number of values τ may obtain at the i-th interaction is k^i.

In order to overcome the above problem we propose a Monte-Carlo-based computational approach. According to this method, when reaching the i-th interaction in the backward induction process, we draw a large set of random sequences of values from the underlying interaction-quality probability function in order to represent with each sequence one possible random flow of experiences if delaying the rating request. We then set the value of T_i to some arbitrary value and emulate the process of determining the interaction at which the rating request will be issued, given the drawn sequence and the thresholds $T_{i+1}, ..., T_N$ (that were calculated in prior stages), assuming $\tau = T_i$. Taking the average of the measure τ at the time of rating request in the emulated processes over all sets obtains an accurate estimate for $E^{\tau}_i(T_i)$. Since the optimal T_i satisfies $T_i = E^{\tau}_i(T_i)$ we can use the difference between T_i and $E^{\tau}_i(T_i)$ as an indicator for whether the optimal T_i value should be higher or lower than the current T_i value. Now we can change T_i accordingly and repeat the process, until the absolute difference between T_i and $E^{\tau}_i(T_i)$ is smaller than some pre-defined accuracy level ϵ.

4 EXPERIMENTAL EVALUATION

In order to evaluate the effectiveness of timing rating requests using ESB, we developed a framework that uses a "virtual investor" in a game called "Investments Game" (implemented using *ASP.NET*). The virtual investor needs to make investments in stocks and mutual funds on behalf of the user. Investment decisions are made sequentially, such that on each day there is a fixed amount of money the investor can invest in one of the alternatives available to it (for 20 consecutive "days"). The return for investing in each alternative is uncertain, though the investor is acquainted with the underlying return probability function (e.g., based on past experience and market conditions). After making an investment, the actual return is revealed both to the virtual investor and the user. The user's utility is linear in the average return obtained. Therefore an optimal decision from a (fully rational) user's point of view should be always picking the alternative associated with the highest expected return.

Participants received a fixed payment for their participation, and a bonus, linearly correlated with the total quality of outcomes obtained, which was the more significant portion of the total payment received. At the end of each period, after observing the quality achieved, the virtual investor had a chance to request the user to rate their performance. Rating was issued based on a ten star scale, 1 being the worst rating and 10 the best. There was no default rating, and the user could not proceed in the game without submitting a rating. Participants were recruited and interacted with through the crowd-sourcing framework of Amazon Mechanical Turk (AMT).

We used two experimental treatments, differing in the method according to which rating requests were timed:

Random - randomly picking for each experiment the round at which a rating request will be issued. This treatment was used as a baseline for comparison due to its wide use.

ESB - extracting the rating request decision thresholds according to ESB as described above. The Pearson correlation was used for determining the proper α value.

Overall, we had 600 participants taking part in the Investments Game experiment. Statistical significance is calculated, whenever applicable, based both on the one-way Mann-Whitney U test and t-test, taking the worse of the two as a matter of precaution.

4.1 Average Rating

The average rating received for the virtual investor using ESB was 7.46, while using Random timing requests resulted in rating of 6.87. The results are statistically significant ($p < 0.05$), meaning that the modeling of user satisfaction through considering the qualities of all prior interactions the user had with the system encompasses a significant contribution to the improvement achieved with ESB.

5 DISCUSSION AND CONCLUSIONS

The encouraging results reported in the paper provide strong evidence for the ability to influence a user's rating through intelligent timing of rating requests. In particular, the use of the proposed ESB method for timing requests resulted in a significantly better average rating compared to when using random rating requests, which is de-facto the standard in many rating systems. The model used in this research relies on several assumptions that do not hold for all application domains. This leaves much room for future research in the form of studying various model variants that are more applicable for specific real-life applications. For example, the modeling of the effect of user refusal to provide rating over the results of future requests and their optimal timing or enabling the agent to override ratings with new ones requested from the user.

REFERENCES

[1] Nadav Altshuler and David Sarne. 2018. Modeling Assistant's Autonomy Constraints as a Means for Improving Autonomous Assistant-Agent Design. In *Proc. of AAMAS*.

[2] Amos Azaria, Sarit Kraus, Claudia Goldman, and Omer Tsimhoni. 2014. Advice provision for energy saving in automobile climate control systems. In *Proc. of AAMAS*. 1391–1392.

[3] Amos Azaria, Zinovi Rabinovich, Sarit Kraus, Claudia V Goldman, and Ya'akov Gal. 2012. Strategic advice provision in repeated human-agent interactions. In *Proc. of AAAI*. 1522–1528.

[4] Nick Babich. 2016. Mobile App UX Design: Prompting For App Review. https://uxplanet.org/mobile-app-ux-design-prompting-for-app-review-5b75c0005cab. (2016). [Online; accessed 1-August-2017].

[5] Neil A Carrier, George S Howard, and William G Miller. 1974. Course evaluation: when? *J. of Educational Psych.* 66, 4 (1974), 609.

[6] Gang-Len Chang and Hani S Mahmassani. 1988. Travel time prediction and departure time adjustment behavior dynamics in a congested traffic system. *Transportation Research Part B: Methodological* 22, 3 (1988), 217–232.

[7] Avshalom Elmalech, David Sarne, Avi Rosenfeld, and Eden Shalom Erez. 2015. When Suboptimal Rules.. In *AAAI*. 1313–1319.

[8] Kenneth A Feldman. 1979. The significance of circumstances for college students' ratings of their teachers and courses. *Research in Higher Education* 10, 2 (1979), 149–172.

[9] Peter W Frey. 1976. Validity of student instructional ratings: Does timing matter? *The Journal of Higher Education* (1976), 327–336.

[10] Matt Galligan. 2014. The right way to ask users to review your app. https://medium.com/circa/the-right-way-to-ask-users-to-review-your-app-9a32fd604fca. (2014). [Online; accessed 1-August-2017].

[11] Everette S Gardner. 1985. Exponential smoothing: The state of the art. *Journal of forecasting* 4, 1 (1985), 1–28.

[12] Mark Harman, Yue Jia, and Yuanyuan Zhang. 2012. App store mining and analysis: MSR for app stores. In *Proc. of MSR*. 108–111.

[13] Ryan Harter. 2014. Kindly Asking for Ratings and Reviews. http://ryanharter.com/blog/2014/09/05/kindly-asking-for-ratings-and-reviews/. (2014). [Online; accessed 1-August-2017].

[14] Tobias Hoßfeld, Sebastian Biedermann, Raimund Schatz, Alexander Platzer, Sebastian Egger, and Markus Fiedler. 2011. The memory effect and its implications on Web QoE modeling. In *Proc. of ITC*. 103–110.

[15] J Stuart Hunter et al. 1986. The exponentially weighted moving average. *J. Quality Technol.* 18, 4 (1986), 203–210.

[16] Caroline Lee. 2014. 8 Tips to improve app store rating. https://polljoy.com/blog/improve-app-store-rating. (2014). [Online; accessed 1-August-2017].

[17] Priel Levy and David Sarne. 2016. Intelligent Advice Provisioning for Repeated Interaction.. In *Proc. of AAAI*. 842–849.