Deep learning for simulation-based Bayesian inference of hidden parameters in online reputation systems

SHRABASTEE BANERJEE, Tilburg School of Economics and Management, The Netherlands
NARENDRA MUKHERJEE, Tripadvisor LLC, USA
M. AMIN RAHIMIAN, University of Pittsburgh, Dept. of Industrial Engineering, USA

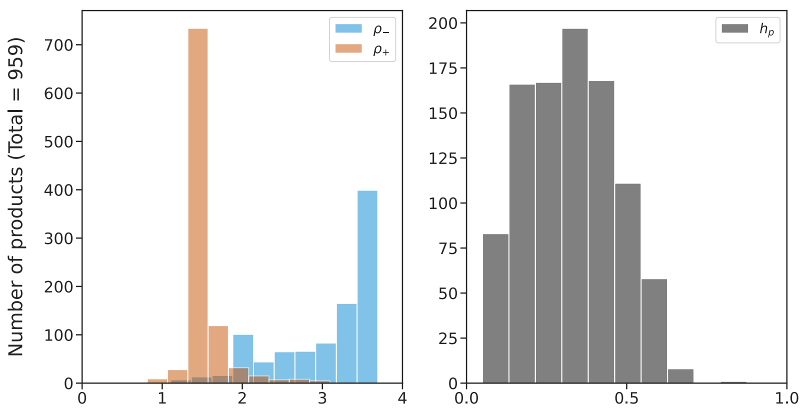

Fig. 1. The distribution of mean values of the marginal posterior distributions of the thresholds to leave a rating (ρ_-, ρ_+) and the product-specific propensity to herd h_p for 959 products on Amazon.com. We see that costs to reviewing are asymmetric (the cost to leaving a negative rating, ρ_-, is larger than that for leaving a positive rating, ρ_+) and the herding probability h_p is significant for the majority of the products, indicating bandwagon effects in rating behavior.

Online reputation systems are an essential component of electronic commerce platforms. However, despite their prevalence, online ratings are subject to selection biases since the decision to leave a rating depends on the specific consumer and their circumstances. There are a number of hidden parameters governing such selection biases but it is difficult to infer them directly from observed ratings given the complexity of reputation systems. In this work, we first propose a generative model that accounts for various behavioral phenomena behind online rating generation (e.g., cost to leaving a rating or herding). We then build upon recent advances in likelihood-free/simulation-based Bayesian inference using deep learning to infer the hidden parameters of the generative model in a scalable manner. The inference engine only takes the time series of ratings as input, and therefore can be used to model correlations of inferred cost parameters with various product features. As a preliminary proof of concept, we apply our model to a dataset of 450,000 product reviews submitted on Amazon.com. We find that the cost to leaving a negative review is much greater than a

Authors' addresses: Shrabastee Banerjee, shrabastee.b@tilburguniversity.edu, Tilburg School of Economics and Management, Tilburg, The Netherlands; Narendra Mukherjee, narendra.mukherjee@gmail.com, Tripadvisor LLC, Needham, Massachusetts, USA, 02494; M. Amin Rahimian, rahimian@pitt.edu, University of Pittsburgh, Dept. of Industrial Engineering, 1006 Benedum Hall, Pittsburgh, Pennsylvania, USA, 15261.

Manuscript submitted to ACM

positive review, and a baseline level of bandwagon effects (in the form of herding) are present for the majority of products. Gaining a better understanding of the dynamics of reputation systems, namely, the conditions under which ratings are submitted, is crucial for marketers, brand managers, and designers of digital platforms, who can leverage this information to stimulate further reviews and better manage user generated content. Working code is available on Github and the docker image containing the dependencies for the code is on Dockerhub.

CCS Concepts: • **Information systems → Electronic commerce**; • **Computing methodologies → Simulation types and techniques**; **Neural networks**.

Additional Key Words and Phrases: Bayesian inference, online reviews, herding

ACM Reference Format:
Shrabastee Banerjee, Narendra Mukherjee, and M. Amin Rahimian. 2021. Deep learning for simulation-based Bayesian inference of hidden parameters in online reputation systems. 1, 1 (January 2021), 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Reviews and ratings have been ubiquitously acknowledged as an influential source of information for prospective consumers on e-commerce platforms. However, the biased nature of ratings has also been well-documented. As shown by many empirical studies, not all consumers write online product reviews due to a variety of reasons such as limited time and effort, differing satisfaction levels, etc [3, 6]. They may also be influenced by the behavior of other customers on the platform, resulting in herding [7]. Given the complex behavioral mechanisms that govern review generation, it is almost impossible to write down a closed-form likelihood $p(x|\theta)$ of review data x given parameters θ that describe the reviewing behavior of consumers. In this work, we focus on two such sets of parameters that are of substantive interest to marketers: a) the "threshold" to leaving a review, i.e., how much the consumer's expected quality should differ from their realized experience to induce them to leave a review, and b) how much herding behavior is exhibited by users buying and rating different products on an online platform. The existing literature does not provide much guidance about what the rating threshold should be in order to explain the observed shape of a given rating distribution, or how this threshold parameter interacts with users' herding behavior for different products. The ability to recover these parameters objectively from observed data would have substantial practical implications by helping platforms and businesses identify groups of products for which review solicitation interventions might be more effective and necessary.

To this end, we use recent advances in likelihood-free Bayesian inference with deep learning [2, 4] to perform posterior estimation in a simulation-based probabilistic model of rating generation. We can then use the fitted posterior estimation neural network to back out cost and herding parameters characterizing the rating distribution of a given product. We apply this approach to a dataset of 959 products (and a total of ∼ 450,000 reviews) and find asymmetrically distributed threshold parameters — the "cost" to leaving a positive review is much lower than that of a negative review. We further find that consumers tend to exhibit some degree of social learning and bandwagon behavior when leaving ratings.

Classical approaches to likelihood-free statistical inference, also known as Approximate Bayesian Computation (ABC) [10] do not scale to high-dimensional applications, and typically rely on ad-hoc choices to design summary statistics and distance functions. In contrast, neural likelihood-free inference is scalable, does not require hand-tuning sample rejection rules as is common in ABC, and is able to handle variable length rating timeseries using an appropriate embedding network. It reliably recovers observed histograms of simulated ratings with the inferred parameters and is orders of magnitude faster than ABC. Moreover, our simulation of rating generation does not use any product features and the fitted posterior inference network (the so-called "inference engine") only takes in the timeseries of ratings as input. As a result, we can use posterior samples drawn from the inference network to uncover interesting post-hoc relationships between the cost parameters and product features such as price and brand (this can be extended to multiple category-specific features). This approach is flexible enough to incorporate and test for other behavioral mechanisms behind review generation that are not already included in our generative model.

2 GENERATIVE MODEL TO SIMULATE ONLINE RATINGS

The first exercise we undertake is to construct a simulation-based probabilistic model of a single customer's journey as they decide on whether and how to rate their purchased product. The model allows us to analyze how such rating decisions affect the overall evolution of ratings. It further offers us a framework via which we can estimate the threshold parameters of interest (i.e, the costs ρ_+ and ρ_- associated with leaving a positive and negative review for the product respectively), and the product-specific propensity to herd h_p. Note that our simulation model also contains a user-specific propensity to herd, h_u; this parameter, however, is a 'nuisance parameter' during the simulations, as the generative model is at the product level. In other words, each simulation corresponds to one h_p and several h_u values, with the latter being picked randomly in [0,1] for every user in that simulation. The key elements of our model are presented in Figure 2, with further details in Appendix B.

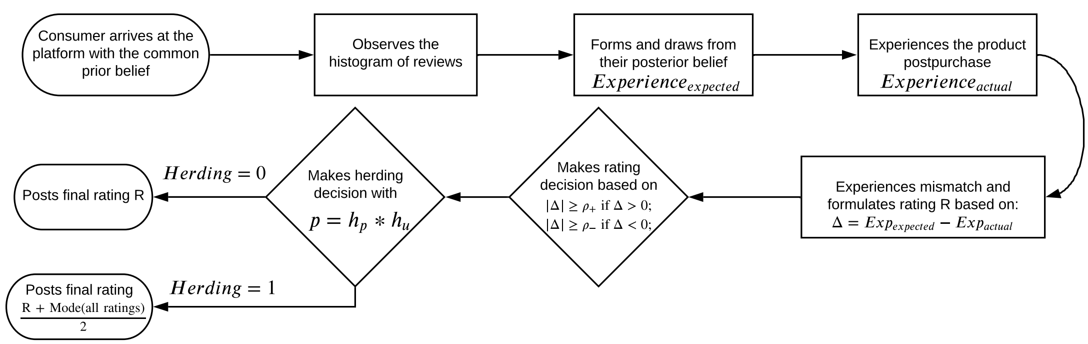

Fig. 2. The generative model governing how consumers leave online ratings on an e-commerce platform.

3 ESTIMATION

We can use the constructed generative model to simulate artificial data given different values of ρ_+, ρ_- and h_p; each (ρ_+, ρ_-, h_p) triplet denotes a different product and can generate a time-series of ratings when passed through the generative model. However, obtaining posterior distributions over the parameters is non-trivial, since the complexity of the data generating mechanism makes it next to impossible to compute the likelihood of data given (ρ_+, ρ_-, h_p). Hence, we rely on state-of-the-art methods of likelihood-free inference, adapting sequential neural posterior estimation (SNPE) to our use case. Specifically, we use the automatic posterior transformation (APT) method implemented in the *sbi* Python package [11]. Although SNPE with APT trains density-estimating neural networks over several rounds, we find in our experiments that a single round of neural network training, with a sufficiently large number of simulations yields posterior inferences that are comparable to multiple rounds of training with a smaller number of simulations. We, therefore, choose to use single round SNPE training as it ensures that the fitted density-estimating neural network is “amortized", i.e, it can be used to infer the joint posterior distribution of (ρ_+, ρ_-, h_p) for any product in the dataset.

We run 20,000 independent simulations of our generative model, wherein for each round, ρ_+, ρ_- and h_p are picked from uniform distributions, along with a randomly chosen total number of reviews picked from Uniform $\in [20, 5000]$. The choice for the total number of simulated reviews is guided by the observed distribution of review volume in our dataset. The range of both ρ parameters is $[0, 4]$ as they correspond to the difference between customers' expected and actual experiences which are on a scale of 1-5; the range h_p, meanwhile, is $[0, 1]$ as it is the probability of herding behavior, averaged across all the customers who rate a given product.

For SNPE, we use a masked autoregressive flow (MAF, [9]) as the density-estimating network, which models complex probability densities by repeatedly transforming a simpler, usually Gaussian, base density. We learn complex patterns in variable length rating timeseries by using a time-dilated 1-D convolutional neural network (CNN) to produce fixed-lenth embeddings that are then fed to the MAF. Dilated 1-D CNNs have been shown to produce reliable and rich encodings of sequential data like timeseries ([8]); we train these sequence embeddings together with the posterior density estimating MAF.

We use the *hyperopt* Python package ([1]) for hyperparameter tuning and neural architecture search. Following architecture search, we use an embedding network that consists of 4 convolutional layers, each with a kernel size of 5 and 8 channels. This is followed by a max pool layer with a kernel size of 5 and 3 fully connected layers that eventually produce a vector of size 32 from each rating timeseries that is passed through the network. Following [8], we use dilation factors of 1, 2, 4 and 8 in the convolutional layers to model long timescale dependencies. Finally, the MAF contains 5 density transforms, each of which is parametrized by a fully-connected layer that contains 50 hidden units.

4 RESULTS

We validate our generative model and inference procedure in several ways. First, we show that histograms of ratings produced by different combinations of cost parameters can be rationalised based on the literature. For example, when $h_p = 0$ and $\rho_- = \rho_+ = 0$, there is no barrier to rating, and we recover a bell-shaped distribution of opinions [6]. On the other hand, when $h_p = 0$ and $\rho_- = \rho_+ = 1$ (equal and positive), we recover a U shaped distribution, consistent with the theory that consumers are more likely to rate extreme experiences (Figure 3 in Appendix A).

Next, we show that we can recover the values supplied in simulations from the trained posterior estimating network (3 representative examples shown in Figure 4 in Appendix A). Note that the “real” posterior is unknown as multiple combinations of the parameters (in addition to the supplied ones) can generate the simulated data which explains the spread of the inferred posteriors. To further quantify the inference network's parameter recovery performance, we repeat this procedure 1000 times with randomly picked values of (ρ_-, ρ_+, h_p): we find that the 95% highest posterior density (HPD) interval of the inferred posteriors contain the true values of ρ_-, ρ_+ and h_p in 94.5%, 95.7% and 94.4% of the simulations respectively. Additionally, to check if the inferred posteriors are “peaked” around the true values supplied during the 1000 simulations, we measure the number of simulations where the posterior places greater than 1/2 probability weight in a narrow band (1/4 of parameter range; 1.0 for ρ_- and ρ_+ and 0.25 for h_p) around each supplied parameter value: we find this to be true in 78.8%, 82.8% and 70.7% of the simulations for ρ_-, ρ_+ and h_p. Thus, not only do the inferred posteriors contain the true parameter values supplied during the simulations but they are also peaked around those true values in the vast majority of cases.

Finally, we turn to real data and infer the latent parameters for 959 products. Our data is collected from Amazon.com [5] under the broad category of Speakers/Headphones, and a total of 452,423 ratings (mean = 471.76 per product, sd = 1011.4). This analysis restricts itself to a relatively small and homogeneous group of products, but given the scalability of our model, we plan to extend our analysis to multiple product groups and categories to uncover interesting heterogeneity. In Figure 1, we find that review costs tend to be asymmetric, with negative ratings requiring more effort that positive ones. We also find evidence that most products tend to exhibit some degree of herding behavior in their ratings - the posterior mean of the herding parameter h_p is also dispersed and indicative of product-level heterogeneity. Note that Figure 1 only displays the means of the marginal posterior distributions of each of the parameters even though the posterior-estimating MAF produces their joint posterior; we plan to use this joint distribution to look at relationships between the parameters in future work.

We are additionally able to analyse product-level heterogeneity in the parameters by correlating them with product features. While it is a simplifying assumption that the generation of product ratings is not governed by product features, it allows us to directly examine how features might be related to ρ_- and ρ_+, without any apriori hypothesis linking the two in the generative model. Our preliminary analyses indicate that products that are priced above their category average and/or belong to top brands (defined by the number of different products being sold across the platform) have higher ρ_- values and lower ρ_+ values, meaning that it is harder to leave a negative rating for these products than a positive rating. This indicates branding and price-based quality signals in rating behavior, a direction we intend to explore more in future work.

5 CONCLUSION

In this project, we apply recent advancements in likelihood-free inference using deep learning to the domain of online user-generated content. We propose a generative model that governs how reviews are posted in an online setting, and recover specific parameter estimates that control the threshold to leaving a rating as well as herding behavior during the rating process. We apply our model to a dataset of $\sim 450,000$ real reviews collected on Amazon.com and find that the cost to review parameters are asymmetric and thus, justify the J-shaped review distribution commonly observed in empirical data. We additionally find evidence for herding behavior that can lead to bandwagon effects while rating for most products on the platform. Our future aim is to expand our inference to multiple product categories, and also to fine-tune our generative model to more accurately capture real world behavior.

REFERENCES

[1] James Bergstra, Brent Komer, Chris Eliasmith, Dan Yamins, and David D Cox. 2015. Hyperopt: a python library for model selection and hyperparameter optimization. *Computational Science & Discovery* 8, 1 (2015), 014008.

[2] Kyle Cranmer, Johann Brehmer, and Gilles Louppe. 2020. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences* (2020).

[3] Chrysanthos Dellarocas and Charles A Wood. 2008. The sound of silence in online feedback: Estimating trading risks in the presence of reporting bias. *Management science* 54, 3 (2008), 460–476.

[4] David Greenberg, Marcel Nonnenmacher, and Jakob Macke. 2019. Automatic posterior transformation for likelihood-free inference. In *International Conference on Machine Learning*. PMLR, 2404–2414.

[5] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*. 507–517.

[6] Nan Hu, Paul A Pavlou, and Jie Zhang. 2017. On Self-Selection Biases in Online Product Reviews. *MIS Quarterly* 41, 2 (2017).

[7] Lev Muchnik, Sinan Aral, and Sean J Taylor. 2013. Social influence bias: A randomized experiment. *Science* 341, 6146 (2013), 647–651.

[8] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499* (2016).

[9] George Papamakarios, Theo Pavlakou, and Iain Murray. 2017. Masked Autoregressive Flow for Density Estimation. In *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc., 2338–2347. https://proceedings.neurips.cc/paper/2017/file/6c1da886822c67822bcf3679d04369fa-Paper.pdf

[10] Scott A Sisson, Yanan Fan, and Mark Beaumont. 2018. *Handbook of approximate Bayesian computation.* CRC Press.

[11] Alvaro Tejero-Cantero, Jan Boelts, Michael Deistler, Jan-Matthis Lueckmann, Conor Durkan, Pedro J. Gonçalves, David S. Greenberg, and Jakob H. Macke. 2020. sbi: A toolkit for simulation-based inference. *Journal of Open Source Software* 5, 52 (2020), 2505. https://doi.org/10.21105/joss.02505

[12] Yi Zhao, Sha Yang, Vishal Narayan, and Ying Zhao. 2013. Modeling consumer learning from online product reviews. *Marketing Science* 32, 1 (2013), 153–169.

A FIGURES FOR VALIDATION OF GENERATIVE MODEL AND INFERENCE PROCEDURE

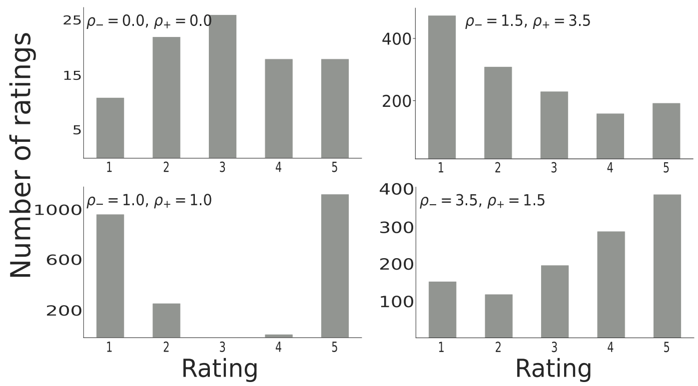

Fig. 3. Different shapes of rating histograms obtained for different combinations of ρ_- and ρ_+ values (herding parameter h_p is set to 0 for these simulations for simplicity). Zero costs lead to a representative (normal) rating distribution, whereas positive and asymmetric costs lead to U and J shaped distributions. Given that our data come from an e-commerce website where positive rating inflation is common, we expect rating distributions and corresponding ρ values to resemble the last panel, but this is a question that we empirically verify.

.

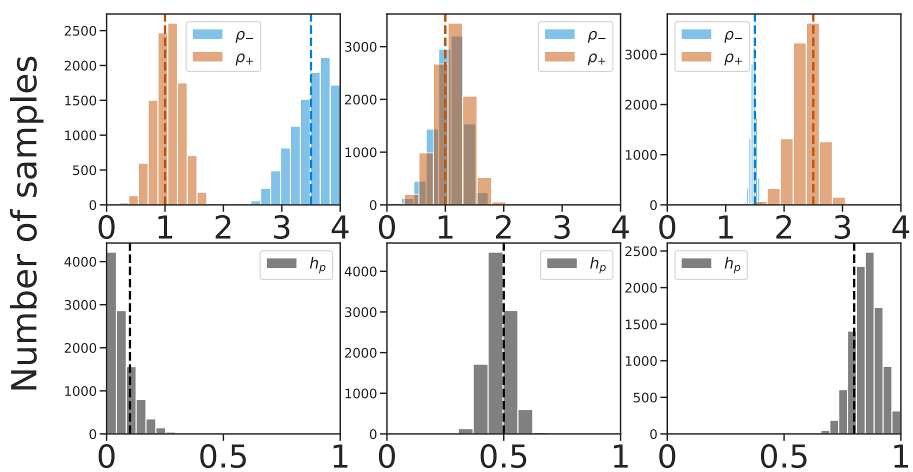

Fig. 4. Samples drawn from the neural posterior density estimator cluster around the true values (dashed vertical lines) of the hidden cost (ρ_+, ρ_-) and herding (h_p) parameters involved in online rating generation. The figure shows parameter recovery for 3 independent simulations from the generative model, one in each column.

B DETAILS OF GENERATIVE MODEL TO SIMULATE ONLINE RATINGS

In this section we present a simulation-based probabilistic model of a single customer's journey as they decide on whether and how to rate their purchased product. The model allows us to analyze how such rating decisions affect the overall evolution of ratings. It further offers us a framework via which we can estimate the threshold parameters of interest (i.e, the cost associated with leaving a review for the product.) It is important to note that in our setup, we do not model how consumers form consideration sets and make purchase decisions. Rather, conditional on purchase, a consumer only decides whether or not to rate the purchased product, and what the rating should be.

Our simulation starts by picking a random number of reviews that needs to be generated and simulating new customers till that number of reviews has been achieved (or, a maximum number of simulated customers has been hit). Each consumer begins with a prior for the review distribution of a product and the current review distribution for the same product. Both of these are modelled by Dirichlet distributions (because Dirichlet is the conjugate prior for the observed multinomial distribution of reviews) as follows:

$$\begin{aligned} \text{Prior} &= \text{Dirichlet}(\alpha_1, .., \alpha_5 = 1.0) \\ \text{Reviews} &= \text{Dirichlet}(\beta_1, .., \beta_5), \end{aligned} \tag{1}$$

where $\alpha_i = 1$ (i.e, a uniform prior), and β_i is the current number of reviews with rating=i, $i \in \{1, 2, 3, 4, 5\}$.

Note that the review distribution prior is the same for each arriving customer and for all products. Next, the customer updates their prior based on the observed distribution (2). This amounts to adding the concentration parameters of the two Dirichlet distributions to form a posterior over product quality:

$$\text{Posterior}_{\text{Quality}} = \text{Dirichlet}(\alpha_1 + \beta_1, .., \alpha_5 + \beta_5) \tag{2}$$

Now, the customer pulls a single multinomial distribution from $\text{Posterior}_{\text{Quality}}$, which yields their "expected" distribution of experiences with the product (presumably combining the product quality with the idiosyncratic fit (random shocks) specific to the customer):

$$\text{Experience}_{\text{Expected}} = \text{Multinomial}(\sum_{i=1}^{5} p_i = 1) \tag{3}$$

where p_i denotes the probability of an experience equivalent to rating i.

Once the customer actually experiences the product, they draw a real "experience" from the multinomial distribution $\text{Experience}_{\text{Expected}}$. This experience is an integer from 1 to 5:

$$\text{Experience}_{\text{Actual}} \in [1, 5] \tag{4}$$

The customer now calculates the difference between $\text{Experience}_{\text{Actual}}$ and the mean of $\text{Experience}_{\text{Expected}}$ as follows:

$$\Delta = \text{Experience}_{\text{Actual}} - \sum_{i=1}^{5} (i \times p_i) \tag{5}$$

This difference determines which rating that the customer assigns to the product (similar to e.g [12]) :

$$\text{Rating} = \begin{cases} 1, & \text{if } \Delta \in (-\infty, -1.5] \\ 2, & \text{if } \Delta \in (-1.5, -0.5] \\ 3, & \text{if } \Delta \in (-0.5, 0.5] \\ 4, & \text{if } \Delta \in (0.5, 1.5] \\ 5, & \text{if } \Delta \in (1.5, \infty) \end{cases} \tag{6}$$

The customer now needs to decide whether to submit their rating to the platform or not. They do so by comparing Δ with the threshold parameter of interest ρ, which is also the object of estimation. Intuitively, small ρ values imply a low cost to rate, and therefore lesser selection bias as more users leave ratings. Given asymmetries in rating distributions, we allow for different ρ values (ρ_+ and ρ_-) for positive and negative Δ respectively. The customer leaves their rating in any of the following situations:

(1) If $\Delta \geq 0$ and $|\Delta| \geq \rho_+$
(2) If $\Delta < 0$ and $|\Delta| \geq \rho_-$
(3) Lastly, we allow for a small, baseline probability of leaving the rating through a “tendency to rate” parameter (set equal to 5% = 0.05), which implies that 5% of consumers might leave a rating even if the above condition is not satisfied. This is done to add an element of stochasticity which is common in online rating environments.

Finally, the rating that the customer has assigned to the product in 6 might be affected by its existing ratings due to herding behavior (leading to bandwagon effects). We instantiate herding behavior in our model through the interaction of 2 different probability parameters: propensities to herd that are specific to the product (h_p) and to the user (h_u). The customer's decision to herd (or not) is, of course, discrete (0 or 1): we postulate that it follows a Bernoulli distribution with a probability parameter $p = h_p \times h_u$. These 2 propensity parameters can then be better understood as:

$$\begin{aligned} h_p &= \frac{\text{Customers that herd while rating product } p}{\text{Total number of customers who rate product } p} \\ h_u &= \frac{\text{Products where user } u \text{ herds while rating}}{\text{Total number of products that user } u \text{ rates}} \end{aligned} \tag{7}$$

Thus, after making the decision to rate, the customer makes a decision to herd with $p = h_p \times h_u$. If they do decide to herd, they modify their determined rating of the product by taking its average with the mode of the product's distribution of existing ratings (we alternatively used the mean/mode of the existing distribution of ratings, but the results were similar). Note that since our generative model is at the product level, we are only able to infer h_p from observed data; h_u is equivalent to a "nuisance parameter" that gets averaged out during the simulations as users with h_u across its range of [0, 1] come and rate a product with a specific h_p.