Detecting and Quantifying Malicious Activity with Simulation-based Inference

Andrew Gambardella [1] **Bogdan State** [2] **Naemullah Khan** [1] **Leo Tsourides** [3] **Philip H. S. Torr** [1]
Atılım Güneş Baydin [1]

Abstract

We propose the use of probabilistic programming techniques to tackle the malicious user identification problem in a recommendation algorithm. Probabilistic programming provides numerous advantages over other techniques, including but not limited to providing a disentangled representation of how malicious users acted under a structured model, as well as allowing for the quantification of damage caused by malicious users. We show experiments in malicious user identification using a model of regular and malicious users interacting with a simple recommendation algorithm, and provide a novel simulation-based measure for quantifying the effects of a user or group of users on its dynamics.

1. Introduction

In 1993 a famous New Yorker cartoon of a computer-browsing canine dryly proclaimed, "on the Internet nobody knows you're a dog." With hindsight this ur-meme has proven prescient with respect to the problem of authenticity on the Internet. That any one Internet user can have identities that are both multitudinous and mutable formed an important part of the network's promise as a medium for communication, self-expression and empowerment. But flexible identities also carry with them the risk of deception, with Internet-facilitated fraud coming to cast a dark shadow over the luminous future originally envisaged by techno-optimists (Friedman & Resnick, 2001).

Stakes increase dramatically when the problem of authenticity meets the power of ranking algorithms, which are responsible for fulfilling and defining information retrieval needs. Tricking an algorithm into honoring at face value those features coming from a certain set of (malicious) users can result in disaster, with unsuspecting users being recommended content, the consumption of which serves purely to enrich a set of attackers rather than to fulfill users' information needs.

Ideally speaking, a good recommendations system should be able to identify and remove malicious users before they can disrupt the ranking system by a significant margin. However, to eliminate the risk of false positives a resilient ranking system can use as much data as possible. So we have to adjust the tradeoff between false positives and the damage a set of malicious users can cause to a ranking system.

Bearing these limitations in mind, as a first approximation, it seems reasonable, for the sake of greater analytical clarity, to divide the user base of an online social network into the vast majority of organic users and a minority of attack profiles. Seen in this way, discussions of inauthentic amplification and coordinated inauthentic behavior fit in with earlier analyses of "shilling" in recommender systems (Si & Li, 2020). Here, we study a popular class of shilling attacks known as profile injection attacks (Williams et al., 2007), in which attackers add bogus accounts to a recommendation system, and attempt to push the ratings of a certain subset of products upward and others downward, while obfuscating their intentions (Ricci et al., 2015).

In our case we are interested in asking the deceptively simple question, *how would ranking outcomes differ in the absence of malicious users*. We define malicious users here as those users misrepresenting either their motives or their identity for strategic gain involving the promotion or demotion of units of content. The question is difficult to answer because of the multitudinous feedback loops potentially at work in recommender systems, which mean that simply counting the effects directly attributable to known malicious users is insufficient for giving an accurate picture of ranking outcomes in the putative counterfactual universe in which no malicious users existed.

Probabilistic programming (van de Meent et al., 2018) has emerged as a principled means of dealing with complex causal scenarios not unlike the issue discussed here, being used in domains as diverse as lion behavior interpretation (Dhir et al., 2017), spacecraft trajectories (Acciarini et al., 2020) and high-energy physics (Baydin et al., 2019a;b).

[1] Department of Engineering Science, University of Oxford [2] scie.nz, work done while at Facebook [3] Facebook Inc.. Correspondence to: Andrew Gambardella <gambs@robots.ox.ac.uk>.

ICML workshop on Socially Responsible Machine Learning, 38^{th} International Conference on Machine Learning, 2021. Copyright 2021 by the author(s).

It is our contention that probabilistic programming, and simulation-based inference (Cranmer et al., 2020) in general, can be used credibly to estimate the difference between realized outcomes and the counter-factual scenario which excludes malicious behavior. We support our assertion by providing:

1. A proof-of-concept detection algorithm, validating that malicious user identification using simulation-based inference techniques is possible using data from the model. This is a necessary but insufficient condition for the computation of a counterfactual scenario.
2. A simulation-based counterfactual measure of influence in a ranking algorithm, grounded in an information theoretical view of the joint probability distributions of ranking outcomes in the presence, and absence, of malicious users.
3. An illustration of how the measure could be applied – we show that malicious users acting in coordination have greater impact on social network dynamics than those acting independently.

2. Related Work

The misuse of recommendation algorithms for adversarial gain dates back to the Internet's first growth spurt as a communication platform, and is inherently intertwined with the history of digital spam, defined as:

> "the attempt to abuse of, or manipulate, a techno-social system by producing and injecting unsolicited, and/or undesired content aimed at steering the behavior of humans or the system itself, at the direct or indirect, immediate or long-term advantage of the spammer(s)." (Ferrara, 2019)

Recommendation algorithms have been a key vector for the amplification of spam since the 1990s, when automated – rather than curated – information retrieval first became feasible in a consumer setting, thanks to Page et al.'s (1999) now-famous development of the PageRank algorithm. Compared to earlier proposals, PageRank notably provided a mechanism which enforced algorithmic resiliency – a recursive definition of popularity which protected against simplistic attempts at faking site popularity for monetary gain through the construction of hyperlinking rings ("spamdexing").

PageRank became the algorithmic foundation for Google, the dominant search engine of the past two decades. Nonetheless, in what would become a common pattern in the Internet industry, its original mechanisms proved only partially effective against adaptive adversaries. The rise of ranking also gave birth to an entire industry, search engine optimization (SEO), dedicated to improving results against the ranking algorithm, sometimes using adversarial "black hat" methods (Malaga, 2010). This evolution, in turn, led to subsequent changes to Google's algorithms to improve their resiliency against adversaries (McCullagh, 2011).

Adversarial attacks against recommendation algorithms have become increasingly prominent recently, given the importance of social media in shaping contentious news cycles rife with misinformation, in particular during the course of events such as the 2016 U.S. general elections (Allcott & Gentzkow, 2017), or the 2018 Brazilian general elections (Machado et al., 2019). Automated posting and engagement, via "social bots" has been recognized as a particularly important factor in the spread of disinformation on social media (Ferrara et al., 2016; Arnaudo, 2017; Shao et al., 2017; Cresci, 2020). The issue of automation is intertwined with that of inauthenticity, with "coordinated inauthentic behavior" (CIB) emerging as a distinct concept both among academics (Giglietto et al., 2020) and industry practitioners (Weedon et al., 2017).

The concept of CIB relies on the existence of "inauthentic accounts," distinguishable from authentic users. The notion of authenticity carries with it a great deal of complexity on the Internet. Our analysis as a result is scoped to those platforms (such as Facebook or Twitter) which rely on the explicit expectation of online identities consistent with offline personas. Even CIB itself should not be seen simply through the binary view of malicious attackers and organic users, as attackers may act to catalyze existing grievances and ideologies present among audiences ripe for manipulation (Starbird et al., 2019).

To fight vote spam in user–item interactions, Bian et al. (2008) proposed training ranking models using a method based on simulated voting spam at training time. Alternatively, Bhattacharjee & Goel (2007) have proposed creating incentives for power users (*"connoisseurs"*) to counteract the influence of spammers. More recently, Basat et al. (2017) have proposed introducing noise into the ranking function to account for distorted incentives leading to the production of low-quality content (e.g., through link farming).

3. Methods

Our examination is meant to provide a minimal example of a recommender system. We choose movie ranking as our setting, given the canonical nature of the task, e.g., IMDb[1] data having a long history of use in the study of recommender systems. This is admittedly a "toy" model, which does not account for more complex designs (i.e., personalization), or for the many issues that intervene in the deployment of rec-

[1] https://www.imdb.com/

ommender systems in the real world (model update cycles, A/B testing, site outages, etc.). Formulating and studying such a model allows us to focus on the derivation of core concepts such as the influence metric (Section 4.2) in the framework of probabilistic programming.

We created a model that represents several malicious users attempting to game a recommendation algorithm modeled as a simple ranking algorithm (without loss of generality, users rating movies and being recommended new movies to watch based on the current mean rating), provided in the supplementary material. In this model, multiple users (some malicious and some organic) are rating items, which are then ranked and suggested to other users based on their ranking. User tastes are modeled by real-valued variables $\nu_i \in [0, 1]$, which determine which movies they would naturally like. Similarly, movies have taste features $\mu_j \in [0, 1]$ which denote something akin to their genre and in our model are left fixed. We model the rating function $\mathrm{Rate}(\upsilon_i, \mu_j)$ so that user i will rate movie j higher the closer user taste ν_i is to movie taste μ_j. The resulting ratings $\rho_{i,j}$ in each user–movie pair constitute the elements of the global rating matrix $\boldsymbol{R}$.

In this model the main latent variables we would like to infer are the binary variables β_i, denoting whether a given user i is malicious or not, and τ_i, denoting the target movie which user i would like to boost, if user i is malicious, i.e., if $\beta_i = 1$. Probabilistic programming will allow us to condition this model on a given rating matrix (for instance, one that represents real-world movie ratings), and then find empirical distributions over the latent variables in the simulator $(\boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{\beta}, \boldsymbol{\tau})$ consistent with the given rating matrix $\boldsymbol{R}_{\text{obs}}$. In summary, for the purposes of identifying malicious users and what they are trying to boost, we will obtain the posterior distribution $p(\boldsymbol{\beta}, \boldsymbol{\tau} | \boldsymbol{R}_{\text{obs}})$, while leaving $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ as nuisance variables.

We implemented our model in PyProb (Baydin et al., 2019b), a lightweight probabilistic programming library for stochastic simulators. We obtain our posteriors using weighted importance sampling (Kitagawa, 1996) which gives a posterior in the form of weighted traces drawn from a proposal distribution, which is in our case the unmodified stochastic simulator. As our posterior is given to us in the form of simulations conditioned on observed data, it is by nature completely disentangled and interpretable, and will tell us the goals of each of the malicious users (τ_i) in addition to their identities (β_i). Crucially, this Bayesian approach also gives us principled uncertainty estimates associated with all our predictions.

4. Experiments

4.1. Obtaining the Posterior and Identifying Malicious Users

We found that obtaining a posterior over the identities of malicious users, as well as their targets, was non-trivial, with different inference engine families behaving considerably differently. Markov-chain Monte Carlo (MCMC) (Metropolis et al., 1953; Hastings, 1970; Wingate et al., 2011), despite being the gold standard to converge to the correct posterior given enough samples, performed extremely poorly. We suspect that this is due to the variables of interest (the malicious users and their targets) making up only a small portion of the latent variables in the model, as well as being discrete whereas the others are continuous. We observed that the model as it is currently formulated was not a good fit for the single-site MCMC (Wingate et al., 2011; van de Meent et al., 2018) inference engine implemented in PyProb, mainly because generating proposals where a single maliciousness latent β_i is flipped lead to very low acceptance probabilities due to the very abrupt nature of the resulting change in the rating matrix (which needs to be compensated by corresponding changes in some user taste latents υ_i that cannot be achieved in a single-site MCMC algorithm), leading to slow mixing and poor sample efficiency.

We found that weighted importance sampling (IS) (Kitagawa, 1996) performed better in practice than MCMC, and used it to obtain posteriors conditioned on observed ratings matrices. When using IS with 100,000 executions of our model in which malicious users do not attempt to disguise their activities (i.e., difficulty $\alpha = 0$), the mean of rating matrices in the posterior, i.e., the posterior predictive $p(\boldsymbol{R}|\boldsymbol{R}_{\text{obs}})$, appears to be a noisy version of the observed ground truth rating matrix $\boldsymbol{R}_{\text{obs}}$, showing that the inference scheme sampled a posterior distribution over simulation runs in which the observed rating matrix is likely.

Much more interesting are scenarios in which malicious users attempt to disguise their activities through obfuscated attacks. We model this obfuscation by setting the difficulty hyperparameter $\alpha = 0.3$. This obfuscation introduces a significant amount of uncertainty into our results, which is reflected in the detection of malicious users. Whereas in the unambiguous case IS produced an empirical posterior over malicious users and malicious target that matched with the ground truth nearly exactly, in the ambiguous case we see significant uncertainty in the empirical posterior. The results show that the most probable (0.85) explanation of the observed rating matrix is that there are no malicious users, although we also see non-negligible (0.15) probability attached to the scenario where there is one malicious user (which we know to be the ground truth for the observed rating matrix). We also see that the mode of the posterior distribution for the malicious users matches the ground truth

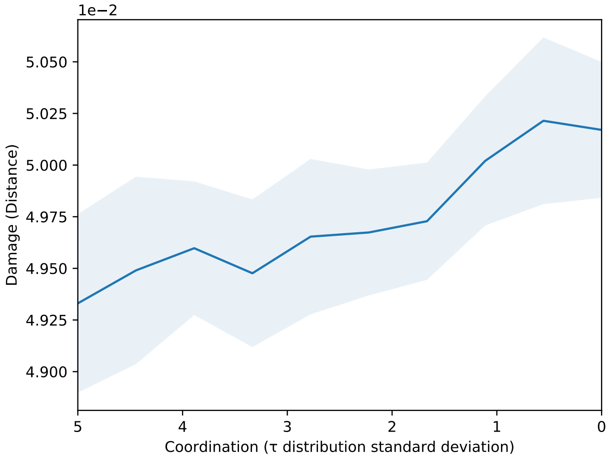

Figure 1. Measuring the effect on the rating matrix as the standard deviation of the distribution from which we draw τ is increased. Lower values of standard deviation (i.e., more coordination between malicious actors) results in a higher impact to the dynamics of the ratings in the matrix. Results averaged over 10 different seeds, with one standard deviation bounds shown.

(user 4), albeit with much lower probability (0.15) compared with the unambiguous case (1.0). We also see that while there is probability mass associated with the ground truth value of the malicious target, this probability is quite low, showing that the IS inference scheme has significant uncertainty in the identification of the malicious target, given the experimental setup and the number of traces (100,000) that we ran during inference.

4.2. Using Counterfactuals to Quantify the Damage Done by Malicious Users

In addition to giving disentangled and interpretable explanations of the behaviour of users interacting with a ranking algorithm, simulation-based inference also gives us the ability to measure the effects of users on the model's dynamics. We can quantify the amount of impact that users imposed on the dynamics of an observed rating matrix by using a slightly modified model, which allows us to disarm users by nullifying the effect of their ratings. Comparing the dynamics reflected in the distributions both with and without disarmed users gives us a counterfactual-based method of quantifying their impact on the dynamics of the simulator. Given a set of disarmed users γ, we find the distance between the posterior-predictive distributions $p(\boldsymbol{R}|\boldsymbol{R}_{\text{obs}})$ and $p(\boldsymbol{R}|\boldsymbol{R}_{\text{obs}}, \gamma)$ given observed data $\boldsymbol{R}_{\text{obs}}$, or between the prior-predictive $p(\boldsymbol{R})$ and $p(\boldsymbol{R}|\gamma)$ in the generic case without an observed $\boldsymbol{R}_{\text{obs}}$.

Our chosen metric for measuring the impact that a disarmed user or group of users has caused is the average JS distance (Endres & Schindelin, 2003), the square root of the symmetric JS divergence (Dagan et al., 1997), computed as the average of JS distances between the counterfactual and real probability distributions over ratings, per entry in the rating matrix. In a distribution over rating matrices $p(\boldsymbol{R}|\cdot)$, each entry in the matrix is a probability distribution (in the empirical case, a histogram) over ratings for each user–movie pair over a large number of simulator executions. Our measure for impact is then the average JS distance for each of these histograms, between the realized and counterfactual rating matrices. The JS distance is a valid metric between probability distributions and always normalised to $[0, 1]$, making it an attractive choice to measure distances between distributions over matrix entries.

We show results using our influence measure between counterfactual and realised ratings matrices while varying the amount of coordination between malicious users in Figure 1. When malicious targets are drawn from a distribution with a low standard deviation, malicious actors act in a more coordinated fashion (as they are more likely to target the same movie), which leads to a higher average distance between the counterfactual and realized ratings distributions. As the malicious target standard deviation increases, malicious users act against each others' interests, leading to a lower overall impact on the dynamics of the model.

5. Discussion

We have suggested the use of probabilistic programming techniques to both discover and measure the influence of malicious users interacting with a ranking algorithm. We base our choice of this method on its conceptual advantages in modeling the full universe of possibilities deriving from the complex interactions between users, content and ranking algorithms. The use of simulation-based approaches in this setting has been limited by practical concerns, until recent advances in numerical computation – coupled with the emergence of high-quality libraries for probabilistic programming.

Probabilistic programming techniques also have some important additional advantages over other methods. They provide for interpretable explanations as to, e.g., why a user would be classified as malicious, and provide measures of confidence in their predictions. Furthermore, generative modeling of the entities and processes involved in this setting allows us to make precise definitions of the core concepts and to quantify key aspects such as user influence and malicious user damage.

References

Acciarini, G., Pinto, F., Metz, S., Boufelja, S., Kaczmarek, S., Merz, K., Martinez-Heras, J. A., Letizia, F., Bridges, C., and Baydin, A. G. Spacecraft Collision Risk Assessment with Probabilistic Programming. In *Third Workshop on Machine Learning and the Physical Sciences (NeurIPS 2020), Vancouver, Canada*, 2020.

Allcott, H. and Gentzkow, M. Social media and fake news in the 2016 election. *Journal of Economic Perspectives*, 31(2):211–236, 2017. ISSN 08953309. doi: 10.1257/jep.31.2.211.

Arnaudo, D. Computational Propaganda in Brazil: Social Bots during Elections. *Computational Propaganda Research Project*, 8:1–39, 2017.

Basat, R. B., Tennenholtz, M., and Kurland, O. A game theoretic analysis of the adversarial retrieval setting. *Journal of Artificial Intelligence Research*, 60:1127–1164, 2017. ISSN 10769757. doi: 10.1613/jair.5547.

Baydin, A. G., Heinrich, L., Bhimji, W., Gram-Hansen, B., Louppe, G., Shao, L., Prabhat, P., Cranmer, K., and Wood, F. Efficient probabilistic inference in the quest for physics beyond the standard model. In *Advances in Neural Information Processing Systems*, volume 33, 2019a.

Baydin, A. G., Shao, L., Bhimji, W., Heinrich, L., Meadows, L., Liu, J., Munk, A., Naderiparizi, S., Gram-Hansen, B., Louppe, G., Ma, M., Zhao, X., Torr, P., Lee, V., Cranmer, K., Prabhat, and Wood, F. Etalumis: Bringing probabilistic programming to scientific simulators at scale. In *International Conference for High Performance Computing, Networking, Storage and Analysis, SC*, 2019b. ISBN 9781450362290. doi: 10.1145/3295500.3356180.

Bhattacharjee, R. and Goel, A. Algorithms and incentives for robust ranking. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, volume 07-09-Janu, pp. 425–433, 2007. ISBN 9780898716245.

Bian, J., Agichtein, E., Liu, Y., and Zha, H. A few bad votes too many? Towards robust ranking in social media. In *AIRWeb 2008 - Proceedings of the 4th International Workshop on Adversarial Information Retrieval on the Web*, pp. 53–60, 2008. ISBN 9781605581590. doi: 10.1145/1451983.1451997.

Cranmer, K., Brehmer, J., and Louppe, G. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062, 2020. ISSN 0027-8424. doi: 10.1073/pnas.1912789117.

Cresci, S. Detecting malicious social bots: Story of a never-ending clash. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 12021 LNCS, pp. 77–88. Springer, 2020. ISBN 9783030396268. doi: 10.1007/978-3-030-39627-5_7.

Dagan, I., Lee, L., and Pereira, F. Similarity-based methods for word sense disambiguation. In *Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pp. 56–63, 1997. doi: 10.3115/979617.979625.

Dhir, N., Wood, F., Vákár, M., Markham, A., Wijers, M., Trethowan, P., Du Preez, B., Loveridge, A., and MacDonald, D. Interpreting lion behaviour with nonparametric probabilistic programs. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2017.

Endres, D. M. and Schindelin, J. E. A new metric for probability distributions. *IEEE Transactions on Information Theory*, 49(7):1858–1860, 2003. ISSN 00189448. doi: 10.1109/TIT.2003.813506.

Ferrara, E. The History of Digital Spam. *Communications of the ACM*, 62(8):82–91, 2019.

Ferrara, E., Varol, O., Davis, C., Menczer, F., and Flammini, A. The rise of social bots. *Communications of the ACM*, 59(7):96–104, 2016. ISSN 15577317. doi: 10.1145/2818717.

Friedman, E. J. and Resnick, P. The social cost of cheap pseudonyms. *Journal of Economics and Management Strategy*, 10(2):173–199, 2001. ISSN 10586407. doi: 10.1162/105864001300122476.

Giglietto, F., Righetti, N., Rossi, L., and Marino, G. It takes a village to manipulate the media: coordinated link sharing behavior during 2018 and 2019 Italian elections. *Information Communication and Society*, 23(6):867–891, 2020. ISSN 14684462. doi: 10.1080/1369118X.2020.1739732.

Hastings, W. K. Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika*, 57 (1):97, 1970. ISSN 00063444. doi: 10.2307/2334940.

Kitagawa, G. Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models. *Journal of Computational and Graphical Statistics*, 1996. ISSN 10618600. doi: 10.2307/1390750.

Machado, C., Kira, B., Narayanan, V., Kollanyi, B., and Howard, P. N. A study of misinformation in whatsapp groups with a focus on the brazilian presidential elections. In *The Web Conference 2019 - Companion of the World Wide Web Conference, WWW 2019*, pp. 1013–1019, 2019. ISBN 9781450366755. doi: 10.1145/3308560.3316738.

Malaga, R. A. Search Engine Optimization—Black and White Hat Approaches. In *Advances in Computers*, volume 78, pp. 1–39. Elsevier, 2010. doi: 10.1016/s0065-2458(10)78001-3.

McCullagh, D. Testing Google's Panda Algorithm: CNET Analysis. https://www.cnet.com/news/testing-googles-panda-algorithm-cnet-analysis/, 2011. URL http://news.cnet.com/8301-31921_3-20054797-281.html.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092, 1953. ISSN 00219606. doi: 10.1063/1.1699114.

Page, L., Brin, S., Motwani, R., and Winograd, T. The PageRank Citation Ranking: Bringing Order to the Web. *World Wide Web Internet And Web Information Systems*, 1998. ISSN 1752-0509. doi: 10.1.1.31.1768.

Ricci, F., Shapira, B., and Rokach, L. *Recommender systems handbook, Second edition*. 2015. ISBN 9781489976376. doi: 10.1007/978-1-4899-7637-6.

Shao, C., Ciampaglia, G. L., Varol, O., Yang, K., Flammini, A., and Menczer, F. The spread of low-credibility content by social bots. *Nature communications*, 9(1):1–9, 2017.

Si, M. and Li, Q. Shilling attacks against collaborative recommender systems: a review. *Artificial Intelligence Review*, 53(1):291–319, 2020. ISSN 15737462. doi: 10.1007/s10462-018-9655-x.

Starbird, K., Arif, A., and Wilson, T. Disinformation as collaborative work: Surfacing the participatory nature of strategic information operations. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):1–26, 2019. ISSN 25730142. doi: 10.1145/3359229.

van de Meent, J.-W., Paige, B., Yang, H., and Wood, F. An Introduction to Probabilistic Programming. *arXiv preprint*, 2018.

Weedon, J., Nuland, W., and Stamos, A. Information Operations and Facebook, 2017. ISSN 1047-9651.

Williams, C. A., Mobasher, B., and Burke, R. Defending recommender systems: Detection of profile injection attacks. *Service Oriented Computing and Applications*, 2007. ISSN 18632386. doi: 10.1007/s11761-007-0013-0.

Wingate, D., Stuhlmüller, A., and Goodman, N. D. Lightweight implementations of probabilistic programming languages via transformational compilation. In *Journal of Machine Learning Research*, volume 15, pp. 770–778, 2011.