# Improving Markov Chain Monte Carlo Estimation with Agent-Based Models

Rahmatollah Beheshti and Gita Sukthankar

Department of EECS
University of Central Florida
Orlando, Florida 32816
{beheshti@knights, gitars@eecs}.ucf.edu

**Abstract.** The Markov Chain Monte Carlo (MCMC) family of methods form a valuable part of the toolbox of social modeling and prediction techniques, enabling modelers to generate samples and summary statistics of a population of interest with minimal information. It has been used successfully to model changes over time in many types of social systems, including patterns of disease spread, adolescent smoking, and geopolitical conflicts. In MCMC an initial proposal distribution is iteratively refined until it approximates the posterior distribution. However, the selection of the proposal distribution can have a significant impact on model convergence. In this paper, we propose a new hybrid modeling technique in which an agent-based model is used to initialize the proposal distribution of the MCMC simulation. We demonstrate the use of our modeling technique in an urban transportation prediction scenario and show that the hybrid combined model produces more accurate predictions than either of the parent models.

**Keywords:** Markov Chain Monte Carlo, agent-based models

## 1 Introduction

Markov chain Monte Carlo (MCMC) simulation is a simple, easily parallelizable methodology for estimating the summary statistics of a population from minimal information. The aim of the process is to approximate the posterior distribution of the model parameters based on the observed data. By using Monte Carlo simulations to perform the high-dimensional integrations necessary to calculate marginal and posterior distributions, algorithms such as Metropolis-Hastings can make the Bayesian inference process tractable. MCMC has been used as a key component in the model fitting process in many types of social modeling and prediction problems. For instance, Cauchemez et al. use a Bayesian MCMC approach to examine the main characteristics that affect influenza disease transmission between households [1]. Similarly, the effect of spatial influences on geopolitical conflicts has been modeled using an MCMC formulation in which the likelihood of war involvement for each nation is conditioned on the decisions of proximate states [2].

Although the MCMC methodology has many advantages, many of the commonly used MCMC algorithms are strongly dependent upon good initialization of the proposal distribution. In cases where the proposal distribution is far from the desired posterior distribution the algorithm may converge to a poor local minimum or require a long time to achieve convergence. In this paper, we focus on the question of how to select a good proposal distribution for MCMC algorithms. To address this problem, we turn to another modeling technique, agent-based modeling (ABM), to generate simulated data which is then used to initialize the proposal distribution of the MCMC. The combination of the two models, agent-based and MCMC, produces a more accurate result than either of the parent models and facilitates the MCMC convergence. To demonstrate the strengths of this approach, we present a case study on modeling and predicting transportation patterns and parking lot usage on a large university campus.

## 2 Related Work

Markov Chain Monte Carlo describes a family of methods for performing Bayesian inference through stochastic simulations of a Markov process. In the domain of social modeling and prediction, MCMC is well suited for studying the effect of long-term influences on dynamic systems of social agents. For instance, SIENA (Simulation Investigation for Empirical Network Analysis) uses MCMC for analyzing longitudinal data of networks and behavior [3]. SIENA is a powerful toolkit that can be used to test hypotheses about the effects of actor and tie covariates on network structure and actor behavior [4]. However, for large and complicated datasets, it can be challenging to get the MCMC component of SIENA to converge in a reasonable period of time. Since our proposed method initializes the proposal distribution at a point closer to the target distribution, it improves the convergence rate of MCMC.

MCMC is an alternative to two other commonly used approximation methods: 1) importance sampling—samples are drawn from a distribution other than the target one, then reweighted to account for differences between the two distributions, and 2) variational inference—the original integration problem is transformed into an optimization problem [5]. Effectively MCMC allows us to draw samples from a distribution $\pi(x)$ without having to know its normalization. With these samples, it is possible to compute any quantity of interest about the distribution of $x$, such as means, confidence regions, or covariance [6]. In this paper, MCMC is used as a simulation technique, and the sample set used to characterize the posterior distribution is simply compared against the output of other simulation techniques such as agent-based modeling, rather than used to perform Bayesian inference over model parameters.

This paper focuses on improving the performance of the Metropolis-Hastings algorithm (MH) [7] which is relatively sensitive to the initial proposal distribution. It is because of this sensitivity that researchers sometimes opt to use alternative MCMC algorithms, such as Gibbs sampling [8]. Our proposed method is a variation on the idea of using suboptimal inference and learning algorithms to

generate data-driven proposal distributions for the MH algorithm [9]. Eaton et al. [10] used dynamic programming to create a proposal distribution for MCMC in the space of directed acyclic graphs. They showed that this hybrid technique converges to the posterior faster than other methods, resulting in more accurate structure learning of graphical models and higher predictive likelihoods on test data.

In [11], de Freitas et al. introduce two different methods to overcome the problem of finding a good proposal distribution. In the first approach, a mixture of two kernels is used to drive the search process: 1) a variational kernel to broadly explore the problem domain and locate regions of high-probability and 2) a Metropolis kernel to explore the local regions. One drawback with this method is that finding a good variational kernel can be difficult to do. To combat this issue, the authors propose a second technique called adaptive MCMC in which the proposal distribution is updated at run-time based on the behavior of Markov chain. Adaptive methods generally seek to construct a better proposal distribution through the combination of stochastic approximation and MCMC [12]. One issue with this class of adaptive techniques is that they often rely on certain mathematical assumptions being valid, and thus can only be used in a limited set of conditions unlike our proposed approach. Reversible jump MCMC is a different form of run-time modification in which the dimensionality of proposal distribution is changed; this technique can be used even in cases that the number of parameters is not known [13]. Brooks et al. introduced a new methodology for constructing efficient reversible jump MCMC proposal distributions [14].

Agent-based modeling can be an effective way of modeling complex systems that are not easy to characterize analytically. Typically, each agent in the simulation operates according to a set of simple rules representing the decision-making process of a human, or a group of humans. Simulating the social system reveals emergent interactions between the agents, which are often not immediately obvious from the rules of the system. For a more comprehensive overview of agent-based modeling approaches and applications, the reader is referred to [15]. Although agent-based systems are a powerful simulation and modeling tool in the hands of a domain expert, it is generally difficult to reproduce or verify conclusions drawn from more complicated ABMs since it rarely possible to exhaustively describe all the interactions which occur within the ABM or to quantify the impact of software modifications to the simulation. In this paper, since the ABM is used exclusively to shape the proposal distribution, it is easy to quantify the contribution of the ABM and reproduce the results.

## 3 Method

Figure 1 provides an overview of our proposed hybrid modeling technique in which an agent-based model is used to generate the proposal distribution used by the Markov Chain Monte Carlo algorithm. For this paper, we present a case study illustrating the usage of our technique as part of modeling effort to under-

stand transportation patterns and parking lot occupancy on the campus of the University of Central Florida.

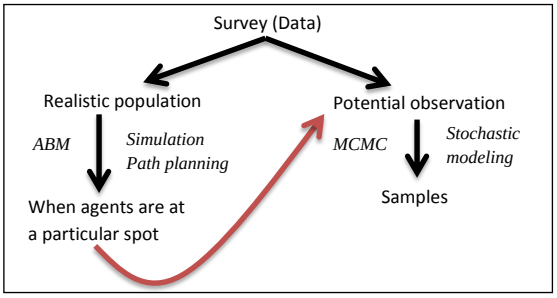



Fig. 1: Overview of proposed method

First, we distributed an online survey to the population of interest on campus-wide email lists; the results of the survey were then used to initialize an ABM model with reasonable parameters. The ABM creates 1) a realistic simulated campus population according to parameters fitted with the survey data 2) schedules for the simulated population members using an activity-based microsimulation 3) paths for the agents to move between their scheduled activities. General trends of student movement can be viewed using the ABM. To estimate a specific quantity of interest, such as usage for a specific parking lot at a particular time and day, the MCMC is used. The Metropolis-Hastings algorithm is initialized using a proposal distribution based directly on the output of the ABM and run to convergence.

In this paper, we compare the prediction performance of three different modeling techniques: 1) ABM only 2) an MCMC using a standard proposal distribution combined with observed data based directly on the surveys 3) our hybrid method in which the classic MCMC sampling is done in two separate phases. In the first phase, data from the agent-based population is sampled to create a proposal distribution, and in the second phase the Markov chain is repeatedly sampled to obtain the target distribution.

### 3.1 Agent-based Transportation Model

To perform transportation forecasting on the UCF campus, we created an agent-based model for simulating the common activities (transportation, dining, recreation, and building occupancy) performed by the 47,000 students on the main campus. 1003 students responded to our online survey posted on `KwikSurveys` which was advertised on various campus email lists. The questions on the survey were grouped into different categories, related to possible places that could be visited on the main campus, and students were specifically queried about

their visitation frequencies. Based on this data, we created the activity-based microsimulation described in [16].

Each agent in the model represents an individual student and has a unique set of parameters that govern his/her activity profile. An agent's defining parameters are: *entrance*, *dormitory*, *department*, *class building*, *arrive*, *depart*, *lunch*, *dinner*, *beverage*, *recreation and wellness*, *parking*, *shuttle*, and *miscellaneous*. The first four parameters designate the single (most common) value of the agents' entry point to the campus, housing situation, home department, and main class building. *Arrive* and *depart* are lists showing the times the agent enters the campus and leaves it. The remaining parameters are lists of locations for the agent's dining, recreation, and commuting. Additionally, each parameter that includes a location has another matching parameter that shows the time or frequency of visiting that location.

Rather than directly mapping the survey data to simulated entities that match the exact preferences of one of the survey respondents, we attempt to learn a general model of the population by fitting a set of distributions to the answers of every question. When the simulation commences, all the agents are initialized with parameters that remain constant over the lifetime of the agent and are used to create daily activity profiles. Our simulation is implemented in the Netlogo [17] environment and is freely available at: `http://code.google.com/p/ucf-abm/`.

### 3.2 MCMC

To benchmark the performance of our ABM MCMC model, we created a Markov Chain Monte Carlo simulation with a standard proposal distribution for making a limited set of forecasts based on the survey data. We use the Metropolis-Hastings algorithm as follows:

- Select a proposal distribution $Q$
- Initialize the starting point, $x_0$
- Do
  - Generate a candidate point $x_c$, according to the probability $Q(x_c|x_i)$
  - Calculate the acceptance probability:

$$\alpha(x_i, x_c) = min(1, \frac{\pi(x_c)q(x_i|x_c)}{\pi(x_i)q(x_c|x_i)})$$

  - Choose $x_{i+1} = x_c$ with probability $\alpha$, $x_{i+1} = x_i$ with probability $(1 - \alpha)$

This procedure is executed until the Markov chain has reached its stationary distribution according to a convergence diagnostic. To validate the simulation, MH is used to estimate the number of cars entering the parking lots at different times of a day. One can envision this as a two dimensional diagram with the horizontal axis corresponding to the time of a day, and the vertical one showing the number of cars entering a specific parking lot. The survey data from the questions about the attendance pattern and frequency of parking lot usage is used to initialize observed data used by the MCMC model. Our MCMC model assumes the unnormalized distribution, $\pi(x)$, is of the form of a Poisson distribution, and a standard multivariate Gaussian is used for the proposal distribution.

### 3.3 ABM MCMC

In our proposed method, the samples produced by the ABM are used to construct the proposal distribution. Then this distribution is employed by the MCMC method to find the target distribution. In this case study, the goal of the campus modeling problem is to build a model describing the transportation patterns of students, hence the distribution that we are seeking (the target distribution) should represent the location of students at different times. The samples that are collected from the agent-based model include $x$ and $y$ coordinates of agents at each hour. This produces a population of samples containing $x$, $y$ and $time$. The proposal probability of each vector is set equal to the number of times the vector exists in the dataset divided by the total number of dataset records. This makes the implicit assumption that the agent-based model has produced an evenly distributed set of samples from the population domain.

## 4 Results

One of the main applications of our microsimulation is analyzing pedestrian movement and car traffic on campus. Figure 2 shows the average visitation frequency for UCF campus locations (junctions, roads, and buildings) as predicted by the ABM MCMC simulation. The darkness of the circles in Figure 2 is proportional to the number of the students who passed or visited these places.

A question of daily interest for most students is parking lot usage: which lots have vacancies and where can the best parking spots be found? UCF Parking Service performed a visual survey of lot usage in Fall 2011 and created a data set which we compared to our hourly forecasts of student lot usage. Figures 3a and 3b show the microsimulation forecasts for the different student parking lots as predicted by: 1) **ABM**: the agent-based model; 2) **MCMC**: the Markov Chain Monte Carlo with standard proposal distribution 3) **ABM MCMC**: the proposed hybrid method. The horizontal axis shows the names of the parking lots and the vertical the difference between the model predictions and the actual parking lot occupancy tallied by UCF Parking Office. The ABM is much better at predicting parking lot usage, compared to the MCMC (standard proposal distribution). However, the hybrid method produces estimations of parking lot usage that are virtually identical to the actual parking lot survey, with improved convergence rates.

## 5 Conclusions

This paper introduces a new hybrid modeling method for combining agent-based models with MCMC. We demonstrate that the proposed method for initializing the MCMC proposal distribution with ABM data significantly reduces the prediction error over standard MCMC and also improves upon the ABM alone. We hypothesize that the combined ABM MCMC finds a more general model of the the posterior distribution than the ABM alone. Although agent-based models

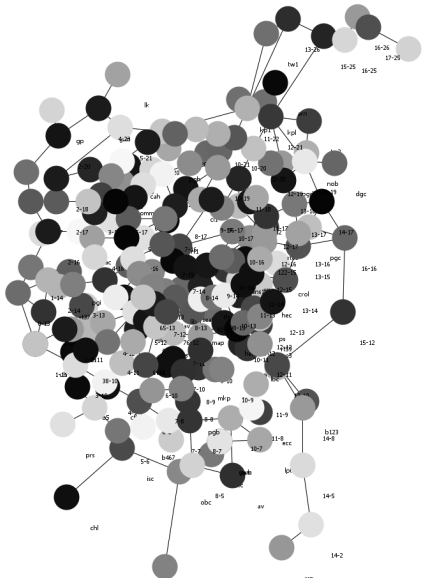
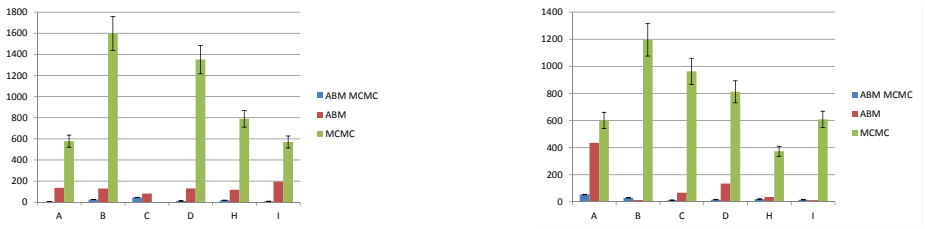


Fig. 2: Average traffic through different locations on the campus as predicted by ABM MCMC estimation with darker circles showing more probable locations. The simulation clearly shows several campus usage trends that are easily verified, including high student union usage (center) and high traffic at main campus entrances (bottom left and up left).




(a) (b)

Fig. 3: The absolute value of the prediction error of the MCMC simulation with standard proposal distribution (**MCMC**), the agent-based modeling method (**ABM**), and our proposed method (**ABM MCMC**). Shorter bars represent predictions that diverge less from the actual observed Parking Services data. Our proposed method accurately forecasts the parking lot usage across all the parking lots (A-I) at noon (3a) and 4 pm (3b).

are often difficult to formally specify and reproduce exactly, the contribution of the ABM can be entirely quantified by the single proposal distribution, which makes it possible to reproduce the results without replicating the entire ABM.

## 6 Acknowledgments

This research was supported in part by NSF award IIS-0845159.

## References

1. Cauchemez, S., Carrat, F., Viboud, C., Valleron, A.J., Bolle, P.Y.: A Bayesian MCMC approach to study transmission of influenza: application to household longitudinal data. Statistics in Medicine **23**(22) (2004) 3469–3487
2. Ward, M.D., Gleditsch, K.S.: Location, location, location: An MCMC approach to modeling the spatial context of war and peace. Political Analysis **10**(3) (2002) 244–260
3. Snijders, T.: Markov Chain Monte Carlo estimation of exponential random graph models. Journal of Social Structure **3** (2002)
4. Snijders, T.: Models and methods in social network analysis. Cambridge University Press, New York (2005)
5. Carbonetto, P., King, M., Hamze, F.: A stochastic approximation method for inference in probabilistic graphical models. In: NIPS. Volume 22. (2009) 216–224
6. Press, W., Teukolsky, S., Vetterling, W., Flannery, B.: Numerical Recipes: The Art of Scientific Computing. Cambridge University Press (2007)
7. Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., Teller, E.: Equation of state calculations by fast computing machines. Journal of Chemical Physics **21** (1953) 1087–1093
8. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Transactions on Pattern Analysis and Machine Intelligence (6) (1984) 721–741
9. Andrieu, C., De Freitas, N., Doucet, A., Jordan, M.: An introduction to MCMC for machine learning. Machine Learning **50**(1) (2003) 5–43
10. Eaton, D., Murphy, K.: Bayesian structure learning using dynamic programming and MCMC. In: Proceedings of the Conference on Uncertainty in Artificial Intelligence. (2007) 101–108
11. De Freitas, N., Højen-Sørensen, P., Jordan, M., Russell, S.: Variational MCMC. In: UAI. (2001) 120–127
12. Andrieu, C., Moulines, É.: On the ergodicity properties of some adaptive MCMC algorithms. The Annals of Applied Probability **16**(3) (2006) 1462–1505
13. Yeh, Y., Yang, L., Watson, M., Goodman, N., Hanrahan, P.: Synthesizing open worlds with constraints using locally annealed reversible jump MCMC. ACM Transactions on Graphics (TOG) **31**(4) (2012) 56:1–56:11
14. Brooks, S., Giudici, P., Roberts, G.: Efficient construction of reversible jump Markov chain Monte Carlo proposal distributions. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **65**(1) (2003) 3–39
15. Macal, C., North, M.: Tutorial on agent-based modelling and simulation. Journal of Simulation **4**(3) (2010) 151–162
16. Beheshti, R., Sukthankar, G.: Extracting agent-based models of human transportation patterns. In: Proceedings of the ASE/IEEE International Conference on Social Informatics, Washington, D.C. (December 2012) 157–164
17. Wilensky, U.: (1999) NetLogo. Evanston, IL: Center for Connected Learning and Computer-Based Modeling, Northwestern University. Retrieved from: http://ccl.northwestern.edu/netlogo/.