BigTop Bazaar: Modeling & Simulating Conference Attendee Dynamics

Ronald J. Nowling[1,2]

[1]Red Hat, Inc. [2]Computer Science & Engineering, University of Notre Dame

Abstract

Where do you get data for building and testing a system for real-time tracking and analysis of attendee movements at a live event like a conference? Simulate it!

We present a physical model for simulating attendees' movements using the Langevin stochastic differential equation (SDE) with booths modeled as attractive sources and the conference space modeled using spherical, harmonic boundary conditions. The strengths of the interactions between booths and attendees are parameterized from a randomly-generated latent factor model.

We analyze the properties of the model, showing the effects of the parameters on the models' dynamics. The resulting time-series data is analyzed to show how the model captures the attendees' preferences through the distributions of the attendees' "residence times" at the booths. Lastly, we discuss how the model can be used to build simulations to support development and testing of analytics software.

Benchmark & Load Testing Use Cases

With advances in mobile technology and internet connectivity, the uses cases for embedding sensors and real-time data collection have increased dramatically [3]. Examples of from the new Internet of Things (IoT) paradigm include health-oriented activity trackers, crowd-sourced traffic monitoring, and crop and soiling monitoring in precision agriculture. With the rise of the IoT, the conversation around Big Data has shifted from "volume" to "velocity."

Synthetic data generators employing domain-specific models have been used successfully for stress testing big data systems [4,5]. We plan to use BigTop Bazaar in a similar manner to generate steaming sensor data to support the development and testing of analytics applications based on frameworks such as Apache Spark, Apache Storm, and Apache Samza under realistic conditions. Metrics such as events per second and processing time per event can aid in capacity planning and optimizing applications for resource usage. By using a model with complex dynamics, we can evaluate machine learning models and balance accuracy with computational performance.

References

[1] C. Gardiner, *Stochastic Methods: A Handbook for the Natural and Social Sciences*, 4th ed. (Springer-Verlag, Berlin, 2009).
[2] C. Masoller, Physics Review Letters, **90**(2):020601, 2003.
[3] M. Chen, et al. Mobile Netw Appl, **19**:171-209, 2014.
[4] R Nowling, et al. IEEE Fourth International Conference on Big Data and Cloud Computing, 2014.
[5] L Gu, et al. IEEE International Conference on Data Engineering, 2015.

Customer Dynamics and Booth Preferences

Drawing from computational chemistry, we model *N* customers as particles under the infuence of *M* booths (attractive sources) in a 2D space. As a simple initial approximation, spherical boundary conditions (SBC), enforced by a harmonic constraint, are used to model the conference space. The dynamics of the particles are modeled by the Langevin stochastic differential equation [1]

$$m\frac{d^2x}{dt^2} = -\nabla U(x) - \gamma m\frac{dx}{dt} + \sqrt{2k_BT}\gamma^{1/2}m^{1/2}dW(t)$$

where with the potential incorporating terms for the booths (attractive sources modeled with scaled Gaussians) and the harmonic spherical boundary conditions:

$$U(x) = \overbrace{\sum_{i,j} -Z_{i,j}\frac{1}{\sigma_j\sqrt{2\pi}}\exp\left(\frac{r_{i,j}^2}{2\sigma_j^2}\right)}^{booths} + \overbrace{\sum_i H(r_{i,c}-r_0)k(r_{i,c}-r_0)^2}^{SBC}$$

$$H(r) = \begin{cases} 0 & \text{if } r \leq 0 \\ 1 & \text{if } r > 0 \end{cases}$$

The interaction strengths $Z_{i,j}$ between customer *i* and booth *j* are parameterized by a latent factor model *Z = LF*. By generating Z in a reduced space, we can ensure that the customers' preferences have a common structural patterns.

The *M*x*K* latent factors *L* and *K*x*N* customer weights *F* matrices are randomly generated. The elements of L and F are randomy sampled from a mixture of Gaussians with means of {-0.75, -0.25, 0.0, 0.25, 0.75}, standard deviations of 0.2, and equal weights. The columns of L are orthogonalized (but not normalized) using the modified Gram-Schmidt process while the columns of F are normalized.

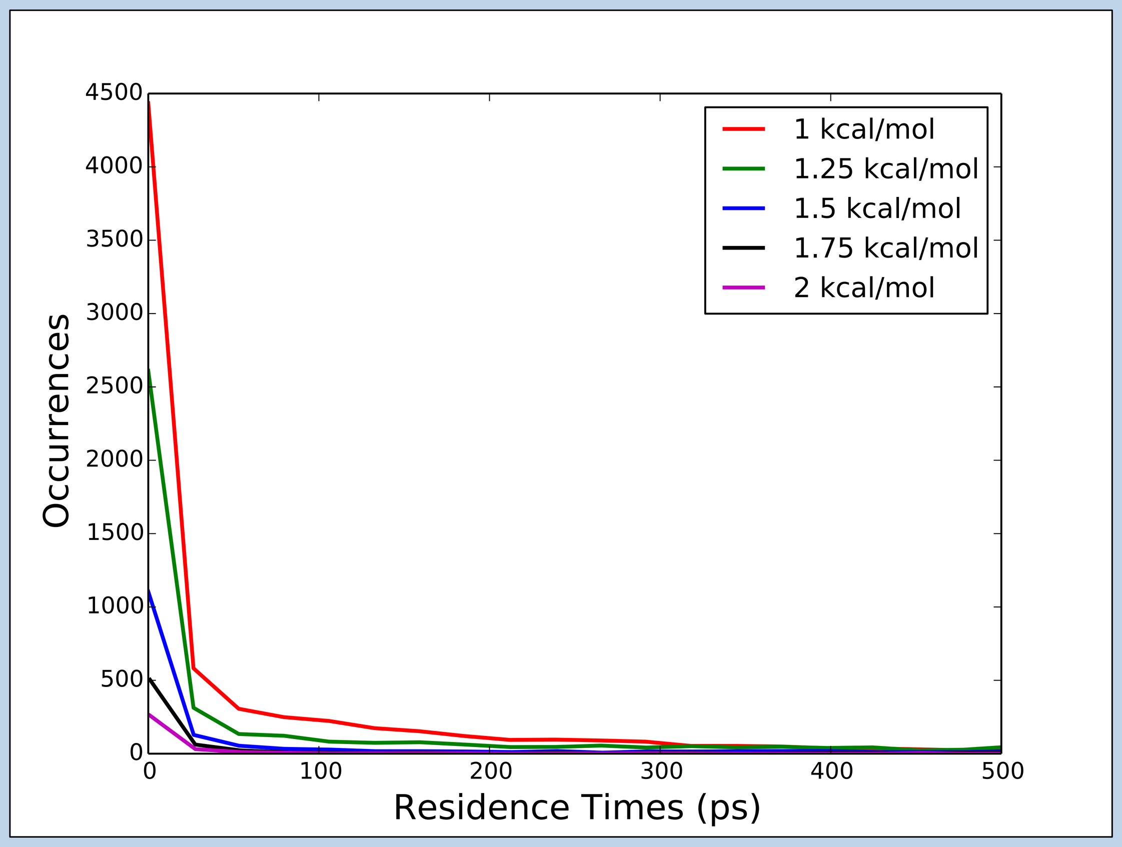
Model Parameters and Variables

N	Number of customers
M	Number of booths
K	Number of latent variables
γ	Damping Coefficient (1/ps)
m	mass (AMUs)
T	temperature (K)
$Z_{i,j}$	Customer-Booth interaction strength (kcal/mol)
σ_j	Booth potential standard deviation
$r_{i,j}$	Customer-Booth distance (nm)
$r_{i,c}$	Customer-Sphere center distance (nm)
r_0	Sphere radius (nm)
k	Strength of harmonic boundary condition (kcal/mol)

Model Properties

We simulated 100 customers (2 AMU masses, damping of 2/ps, and at a temperature of 10° K) with 4 booths (uniform interaction strengths, radii of 1 nm, equally-spaced and separated by 2 nm) for 10,000 ps. The booth interaction strengths were varied from 1 kcal/mol to 2 kcal/mol between runs.

The residence times [2], or elapsed time between when a customer enters and leaves the radius of a booth, were recorded and plotted below. Initial analysis shows that larger interactions strengths result in fewer barrier crossings but similar residence time distributions.



Future work will focus on further validation and analysis of the model. Observed escape and residence times will be compared to those predicted by the literature [1]. Predicting the relationship between the escape and residence times and the interaction strengths, masses, damping, and temperature will provide guidance on tuning parameters to product the desired residence times. Future work will also consider the effects of the latent variable model.

Implementation

BigTop Bazaar is implemented in Java and available from GitHub at `https://github.com/rnowling/bigtop-bazaar` under the Apache Public License v2.

BigTop Bazaar is structured as a library with a small set of classes defining its API. High-level classes correspond to the phases of the model: generating booths, latent variables, customer weights, and simulating particles. Since particles do not interact, simulations can be run in parallel, limiting scaling only by the number of particles and availability of hardware.

The code base is currently in an early stage of developing and undergoing rapid changes. Once stabilized, BigTop Bazaar will be submitted to the Apache BigTop project for inclusion in examples and tests of packaged Big Data frameworks.

Acknowledgements

We would like to thank Will Benton, Andrew Purtell, Casey Robinson, and Jay Vyas for insightful discussion, Red Hat, Inc. for finanical support, and the Apache BigTop project for providing a home for this work and resulting collaborations.