Automated Modeling of Stochastic Reactions with Large Measurement Time-Gaps

Michael Schmidt
Computational Synthesis Lab
Cornell University
Ithaca, NY 14853
mds47@cornell.edu

Hod Lipson
Computational Synthesis Lab
Cornell University
Ithaca, NY 14853
hod.lipson@cornell.edu

ABSTRACT

Many systems, particularly in biology and chemistry, involve the interaction of discrete quantities, such as individual elements or molecules. When the total number of elements in the system is low, the impact of individual reactions becomes non-negligible and modeling requires the simulation of exact sequences of reactions. In this paper, we introduce an algorithm that can infer an exact stochastic reaction model based on sparse measurements of an evolving system of discrete quantities. The algorithm is based on simulating a candidate model to maximize the likelihood of the data. When the likelihood is too small to provide a search gradient, the algorithm uses the distance of the data to the model's estimated distribution. Results show that this method infers stochastic models reliably with both short time gaps between measurements of the system, and long time gaps where the system state has evolved qualitatively far between each measurement. Furthermore, the proposed metric outperforms optimizing on likelihood or distance components alone. Traits measured on the search novelty, age, and bloat suggest that this algorithm scales well to increasingly complex systems.

Categories and Subject Descriptors

I.6.5 [**Simulation and Modeling**]: Model Development – *Modeling methodologies.*

General Terms

Algorithms, Design, Performance.

Keywords

Stochastic modeling, stochastic simulation.

1. INTRODUCTION

Stochastic systems pervade nearly all areas of science, from quantum properties of atomic particles, to chemical reactions in a chemical bath, to fluctuations in populations or ecosystems. All stochastic systems are at least partially random, making them difficult to model dynamically or deterministically. Instead, Monte Carlo methods are often employed to simulate and analyze their behavior.

A particularly important Monte Carlo method was developed by Dan Gillespie in 1977 in order to model chemical reactions kinetics [1]. The Gillespie algorithm performs an exact and statistically-correct simulation of a stochastic system based on a set of discrete chemical reactions, reaction coefficients, and initial conditions. The Gillespie algorithm has been used extensively in systems biology, and also similar domains. Traditionally, the set of reactions that model a stochastic system must be developed and theorized manually by experts.

In this paper we introduce an evolutionary algorithm that automatically hypothesizes about the reactions and reaction rates taking place in a system simply by analyzing raw experimental data, even with large time gaps between observations (see Figure 1). The proposed method searches over a space of reactions in order to find the maximum likelihood model that agrees with the experimental observations.

The key challenge to searching over stochastic models is the computational cost of estimating likelihood values from a model and maintaining a search gradient. Except for only the most trivial systems, the probability density of a set of stochastic reactions cannot be solved over time. Instead, the model can be simulated (or sampled) repeatedly. However, efficient sampling methods fail over large time spans [2], making it difficult to estimate distribution tails.

The proposed method overcomes this difficulty by using a two-component optimization metric. The metric attempts to maximize the log-likelihood of the data given a candidate model. However, if the likelihood is too small to provide a gradient for the search, the criterion changes to the distance of each data point to the estimated probability density of the candidate model. In effect, this distance component allows even extremely inaccurate models to improve despite having zero likelihood. Once models get close enough to the data, where their likelihoods can be estimated accurately through sampling, the metric switches to maximize the likelihood.

This metric also reduces the computational complexity, as the accuracy of estimating the tails of distributions is less important. The algorithm can thereby use fewer samples (fewer simulations of a candidate model) and still estimate a useful likelihood gradient.

2. BACKGROUND

Here we introduce important concepts in stochastic simulation algorithms, density estimation, and evolutionary algorithms.

GECCO'11, July 12–16, 2011, Dublin, Ireland.
Copyright 2011 ACM 978-1-4503-0557-0/11/07...$10.00.

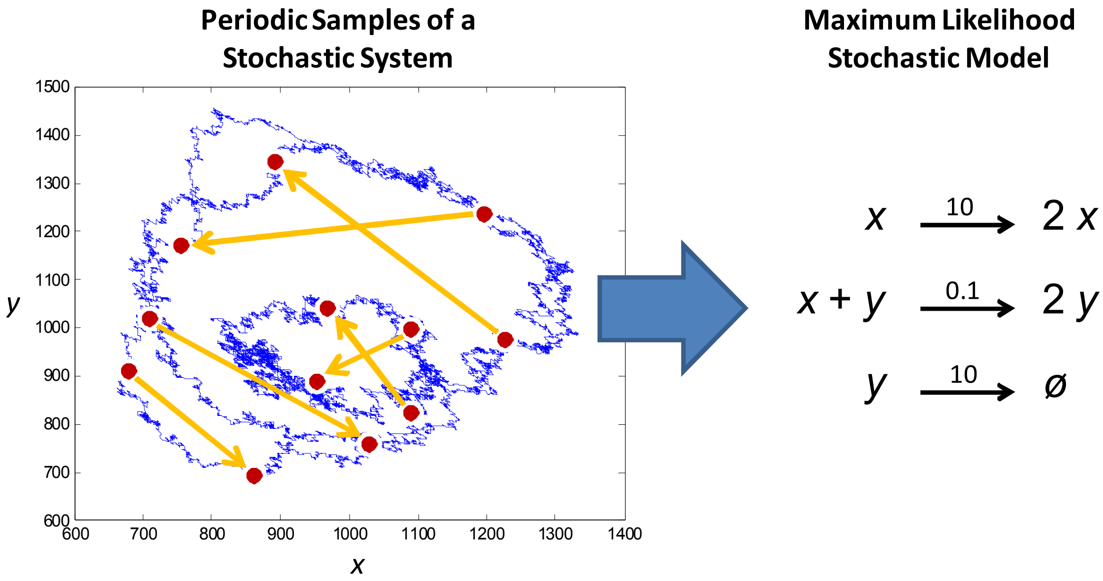

Figure 1. Overview of the modeling problem. A stochastic system evolves an exact behavior over time shown in blue. Periodically, the state of system can be measured (shown in red dots), a sample of the exact time evolution of the system. The task is to infer a maximum likelihood stochastic model (right) for this system from these periodic measurements. Actual data and solution shown.

2.1 Stochastic Simulation Algorithms

The exact stochastic simulation algorithm was first developed in [3] and later applied to chemical kinetics in [1]. The method makes few assumptions about the system except that the environment is well mixed.

The basic algorithm involves two steps: (1) sampling a time delay until the next reaction occurs, and (2) sampling among possible reactions which occurs. Each of these samples are dependent on the number of molecules in the current state. When there are a large number of molecules, the time until the next reaction can be extremely small. The counts of each species also influences which reaction is more likely to occur. The system is simulated by repeatedly applying reactions and incrementing time by the sampled time amount, resulting in a random walk, time-series trajectory. See [1] for more details.

The exact simulation of the Gillespie algorithm becomes critically important when the number of molecules is sufficiently small. In this case, single reactions can significantly impact reaction propensities and future states (e.g. reaching a terminating state). When the number of molecules is exceedingly large, the system dynamics are approximately deterministic because a large numbers of reactions tend to average out random fluctuations.

The exactness of the Gillespie algorithm does come at a computation cost, and several methods have been proposed to improve its performance, while still preserving exactness where necessary.

For our simulations, we use the modified Poisson tau-leaping procedure that ensures that at most one critical reaction occurs per leap [4]. The tau-leaping speeds up the stochastic simulation by estimating the number of reactions occurring during a time period tau. The value of tau is chosen such that the change in reaction propensities during tau is arbitrarily small. When the tau leap is not large enough to provide useful speed up, the algorithm defaults to an exact simulation.

2.2 Kernel Density Estimation

In order to calculate the likelihood of the data given a candidate model, we need to estimate the probability density of the model at each data point. There are many ways to estimate probability densities.

A simple method is to use a histogram. The histogram divides all samples (in our case counts of molecules after simulating a model) into a number of bins. The density is then the bin frequency divided by the bin width. Several methods exist for choosing optimal bin widths and positions [5].

A major drawback to binned histograms however is that they are locally flat everywhere. In other words, they have no local gradient that is amenable to optimization.

An alternative to a histogram, and the method used in our experiments, is kernel density estimation [6, 7]. Kernel density estimation is a non-parametric method to estimate probability density functions. It sums a series of kernel functions that are centered on each sample. We used a Gaussian kernel function, meaning each sample contributed a Gaussian density around its sample value. Choosing a uniform kernel for example would produce a result similar to a binned histogram.

The Gaussian kernel produces density estimates, useful for optimizing, however we still need to specify bandwidth. The bandwidth is analogous to the bin width in a binned histogram. Variable kernel bandwidth selection is the technique of selecting a different bandwidth for each sample [8]. Variable bandwidths allow the kernels to be narrow in high density regions, capturing high details of the distribution, and wide in less certain low-density areas.

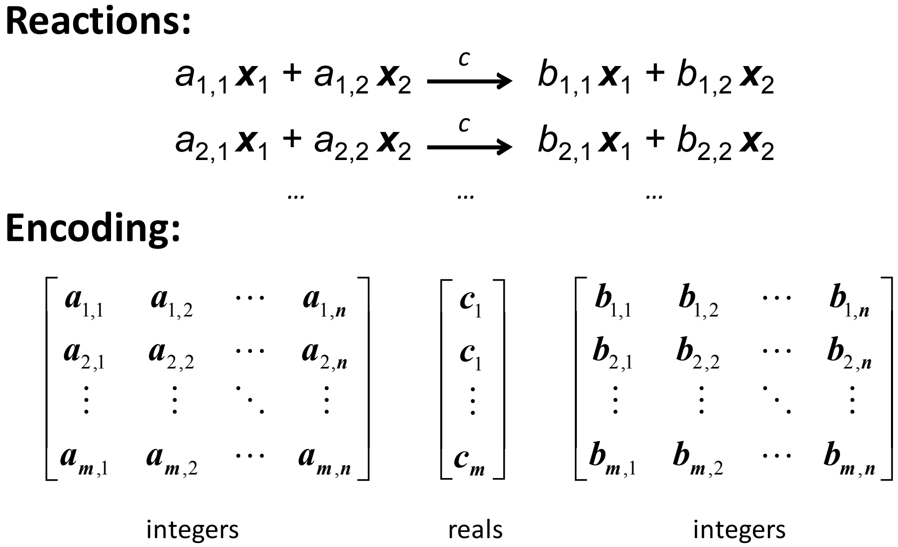
Reactions:
$a_{1,1}\boldsymbol{x}_1 + a_{1,2}\boldsymbol{x}_2 \xrightarrow{c} b_{1,1}\boldsymbol{x}_1 + b_{1,2}\boldsymbol{x}_2$
$a_{2,1}\boldsymbol{x}_1 + a_{2,2}\boldsymbol{x}_2 \xrightarrow{c} b_{2,1}\boldsymbol{x}_1 + b_{2,2}\boldsymbol{x}_2$
... ...
Encoding:
$\begin{bmatrix} \boldsymbol{a}_{1,1} & \boldsymbol{a}_{1,2} & \cdots & \boldsymbol{a}_{1,n} \\ \boldsymbol{a}_{2,1} & \boldsymbol{a}_{2,2} & \cdots & \boldsymbol{a}_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{a}_{m,1} & \boldsymbol{a}_{m,2} & \cdots & \boldsymbol{a}_{m,n} \end{bmatrix}$
$\begin{bmatrix} \boldsymbol{c}_1 \\ \boldsymbol{c}_1 \\ \vdots \\ \boldsymbol{c}_m \end{bmatrix}$
$\begin{bmatrix} \boldsymbol{b}_{1,1} & \boldsymbol{b}_{1,2} & \cdots & \boldsymbol{b}_{1,n} \\ \boldsymbol{b}_{2,1} & \boldsymbol{b}_{2,2} & \cdots & \boldsymbol{b}_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{b}_{m,1} & \boldsymbol{b}_{m,2} & \cdots & \boldsymbol{b}_{m,n} \end{bmatrix}$
integers
reals
integers

Figure 2. The encoding of a solution representing a stochastic model of discrete reactions. A series of chemical reactions (top) are represented by corresponding integer coefficients and real valued rate constants for each reaction (bottom).

In our experiments, we used the square-root law [9] for selecting the bandwidths per sample. This technique requires an initial estimate of the density – here, we used an ordinary histogram with optimize bins chosen by minimizing the mean integrated squared error (MISE) [5]. The final result is a smooth continuous estimate of the probability density that captures both sharp and diffuse features in the distribution.

3. ALGORITHM

The proposed method for inferring a maximum likelihood stochastic model uses an evolutionary algorithm to search for sets of reaction channels and rates to match the data. In this section, we describe the evolutionary encoding of candidate models in the search, and the fitness function.

3.1 Encoding

The stochastic model consists of a series of reactions. Each reaction specifies an integer number for the inputs, an integer number for the outputs, and a real valued number for the reaction rate. If a reaction does not use an input, its input value is 0; likewise for outputs.

We use a fixed, maximum number of reactions for our experiments. Candidate models can opt to use fewer reactions than the maximum by setting the reaction rate to 0, or setting the inputs and outputs to 0.

Figure 2 summarizes our encoding for a stochastic model. It consists of a matrix of integer valued input coefficients for each reaction, a vector of real valued coefficients for each reaction, and a matrix of integer valued output coefficients for each reaction.

A random encoding is produced by filling each matrix with random integers, normally distributed with zero mean and standard deviation of 1, and filling the reaction vector with random positive real values, normally distributed with zero mean and standard deviation of 1.

The mutation operator works by randomizing each individual element with a fixed point mutation probability. The crossover operation recombines two parent encodings to form a new offspring. We use a random single point crossover on the reactions – for example, copying the first n reactions (inputs, outputs, and rate) from the first parent, and the remaining from the second parent.

The complexity of the encoding is defined as the sum of all integer valued reaction coefficients on both inputs and outputs of the reactions.

3.2 Likelihood Estimate

Our goal is to find a maximum likelihood model. We cannot estimated the likelihood of a model explicitly, however, we can estimate the likelihood of seeing the experimental data given a specific model. This gives a measure of how well a particular model agrees with the data. In other words, we are trying to maximize the following expression:

$$Likelihood = \prod_{i=1}^{n} P(x_i \mid M = m)$$

Here, n is the number of data points (measurements of a system state), x_i is a particular data point, m is a particular model, and P is the probability density of the model m at data point i. Rather than working directly with probabilities, it is numerically more stable to work with the log of probabilities, or the Log-Likelihood:

$$Log\ Likelihood = \sum_{i=1}^{n} \log\left(P(x_i \mid M = m)\right)$$

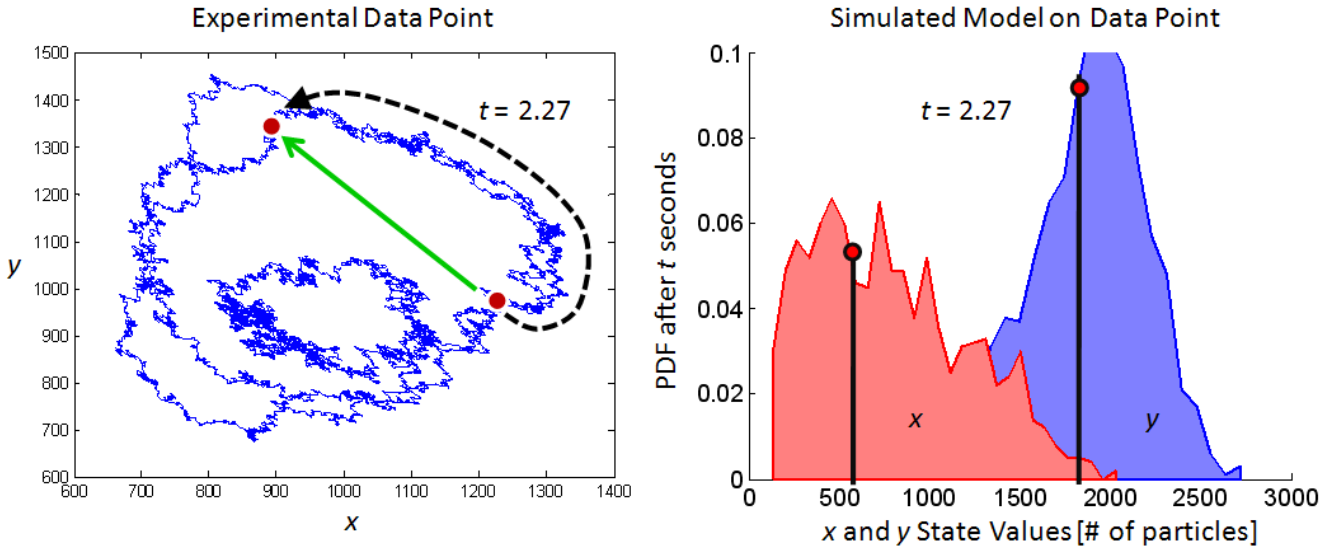

Figure 3. Comparing a candidate model with the experimental data. The left pane shows the hypothetical exact behavior of a system in blue, and two known measurements of the system at red dots. The candidate model is simulated multiple times, starting from the first measurement for *t* seconds, in order to estimate a probability distribution of the model (right). The state of the second measurement is then compared with this distribution to evaluate the quality of the model to reproduce the measurement.

To evaluate the likelihood, we need to estimate the value of $P(x_i \mid M = m)$. We do this by sampling the model m – that is, simulating the model over the time span from the previous data $i - 1$ point to the current data point i.

Figure 3 visualizes the simulation process. The candidate model is simulated, using the previous state, until the time reaches the current state. Each simulation is then added to a kernel density estimator, described above, to estimate the probability density P. The log of the density is then summed for each state x of the system to the cumulative log-likelihood value.

3.3 Fitness Function

Ultimately we want to maximize the likelihood of a candidate model, but since we can only approximate the density function, most random models will tend to have zero likelihood and no gradient to optimize on because we cannot accurately estimate the tails of the probability density function.

Our solution to this problem is to use a two-component fitness metric. The two components are:

1. The log-likelihood as usual, and
2. The distance of the data point to the median value of the estimated distribution

When a model has near zero likelihood (e.g. lower than epsilon = 10^{-6} in our experiments) we subtract the distance of the data point to the median value of the distribution. Otherwise, the fitness is equal to the log-likelihood. This fitness metric is summarized in Figure 3.

By adding the log-likelihood component to the distance component, the fitness function remains monotonically increasing for improving models. This allows initially poor random models to move their distributions close enough to the data points such that their density estimations can be used to maximize the likelihood.

4. EXPERIMENTS

We perform proof of concept experiments on the basic Lotka-Volterra model [10, 11]. The target reactions for this system are shown below:

$$\begin{aligned} x &\xrightarrow{10} 2x \\ x + y &\xrightarrow{0.1} 2y \\ y &\xrightarrow{10} 0 \end{aligned}$$

The Lotka-Volterra reactions model a predator-prey system. In the first reaction, prey (represented by x) grow exponentially. In the second reaction, prey may meet predators (represented by y), causing a prey to die and predators to increase in number. Finally in the last reaction, a predator can die out.

We generated data sets of 10 pairs of measurements of the Lotka-Volterra system. Each pair consists of a random initial condition, followed by a measurement after simulating for a fixed time duration.

In our experiments, we compare two types of data sets, those with short time gaps, where measurements are made in short succession (time steps of 0.002), and long time gaps (time steps of 0.1) where the state of the system changes dramatically between measurements. An example of the long time gaps data set is shown in Figure 3 (left), where each green arrow is a pair of measurements.

In the evolutionary algorithm we use a population size of 30, crossover probability of 50%, and mutation probability of 15%. We allow a maximum of 3 reactions in each model. In estimating a model density for a data point, we sample 100 independent simulations. We track various statistics of the best solution throughout each trial, including fitness on training and test data sets. We terminate all trial runs after 300 iterations (generations) of the evolutionary algorithm.

We repeated multiple trials of the evolutionary algorithm using three different fitness metrics:

1. Log-likelihood only
2. Median distance only
3. The proposed distance and Log-likelihood metric

Therefore, we will be able to evaluate strengths or weaknesses of each component in the proposed metric.

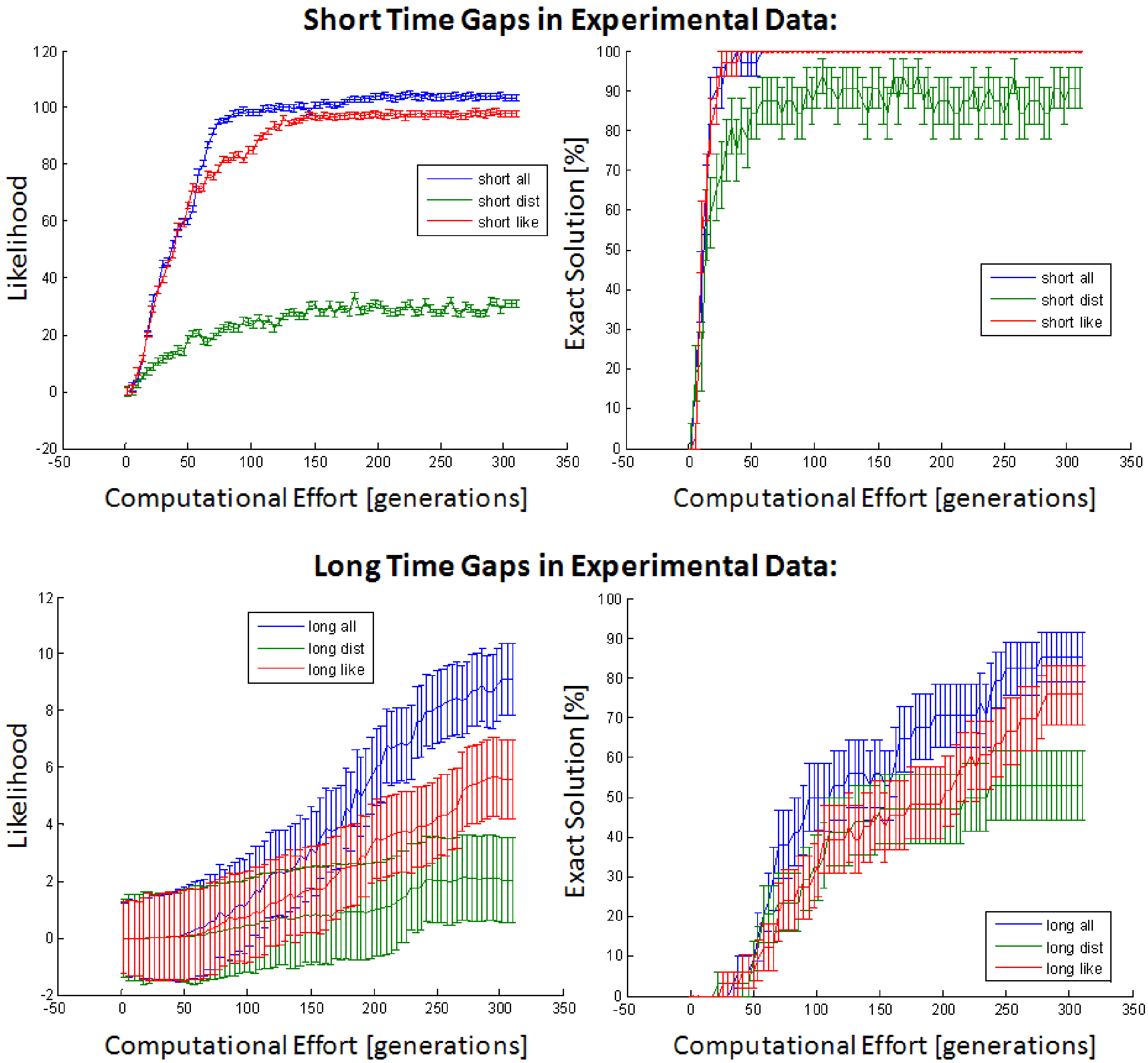

Figure 4. The search performance of the three compared fitness metrics. The top panes show performance when data points appear in rapid succession with short gaps in time. The bottom panes show performance when there are long gaps of time between data points. The left panes show the likelihood score of the best model during the search. The right panes show the percent of runs that identified the exact solution for the amount of computational effort. Error bars indicate the standard error.

5. RESULTS

The first results is that the evolutionary algorithm is able to find the maximum likelihood model for all three compared fitness metrics. For the short time gap data set, Figure 4 (top) shows that all three metrics reach approximately 90% convergence to the exact known model. Both the likelihood and hybrid metrics perform 100% convergence after 100 generations.

In terms of computation time, each generation took approximately 1 minute. Most computation cost lies in simulating various candidate models to estimate their probability densities for each data point.

On the data set with large time gaps, Figure 4 (bottom) shows greater differentiation between the three metrics. The two-component metric reaches the highest likelihood models and convergence, followed by the likelihood only metric. The distance metric only performs the worst.

Interestingly, when the time gaps are short, the performance of the two-component metric and likelihood metric are only approximately similar. This indicates that on short time gaps, the probability density of random candidate models is more likely to provided a useful search gradient, because data points are close to their initial conditions. Here, there is no benefit to using the extra distance component in the fitness metric.

However, the distance metric appears to be crucial when the data set has large time gaps (Figure 4). Here, the two-component metric out performs the other metrics.

Also interesting is that the distance metric alone performs very poorly. This metric allows models to get their distributions centered on the data, but does not optimize the likelihood making it inadequate on its own.

In Figure 5 we compare the relationship between the log-likelihood score and the distance metric. We can see that the distance is correlated with the log-likelihood, but imperfect. There is large variance vertically in the log-likelihood for fixed distance, indicating that log-likelihood metric is inaccurate or at least unstable at the tails of the model probability distribution.

Finally, we collected various traits of the best solution for each algorithm during each search, shown in Figure 6. The first observation is that the genotypic age [12] of the best solution (measured in generations) is roughly equal to the total generations on average. This indicates that the evolutionary search is not being trapped by local optima, otherwise the best solutions would appear younger as younger solutions would replace solutions in local optima. Interestingly, the distance metric algorithm tended to have the highest ages, suggesting that it avoided local optima most, perhaps by identifying an attracting region for the global optima most reliably.

The novelty of the best solution over time, shown in Figure 6, shows that the populations are initially very diverse before converging onto optima. But no clear difference between the compared metrics is apparent. Novelty [13] is defined as the average distance summed over the reaction coefficients of a candidate solution to nearest neighbors in the current population.

In terms of bloat [14], the algorithm starts off with a low bloat ration after random initialization. The bloat tends to increase quickly, and then fall toward a ratio of 1 (no bloat) as the best solution converges to the target (Figure 6). The distance only

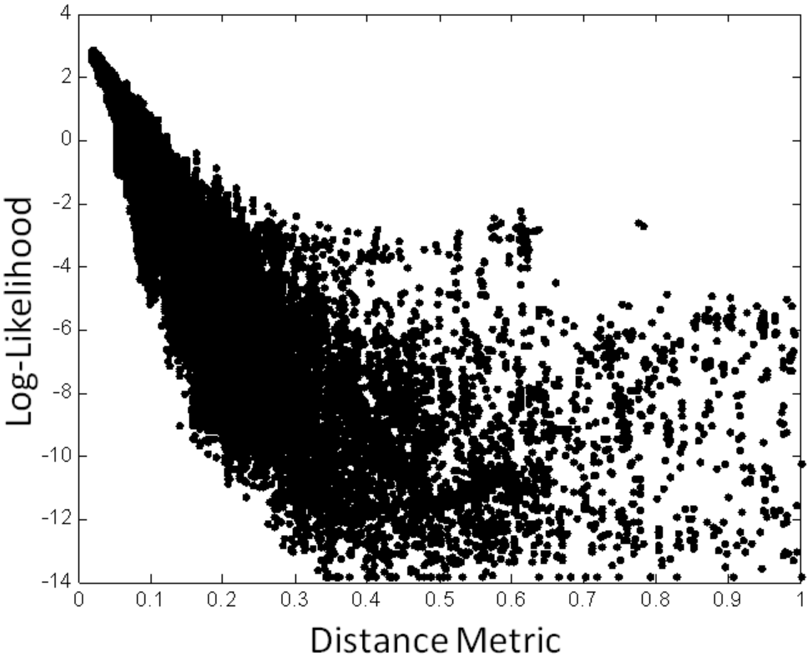

Figure 5. The relationships between the distance metric of a model and its corresponding likelihood given the experimental data. Each point in the plot is a random candidate model during the likelihood search.

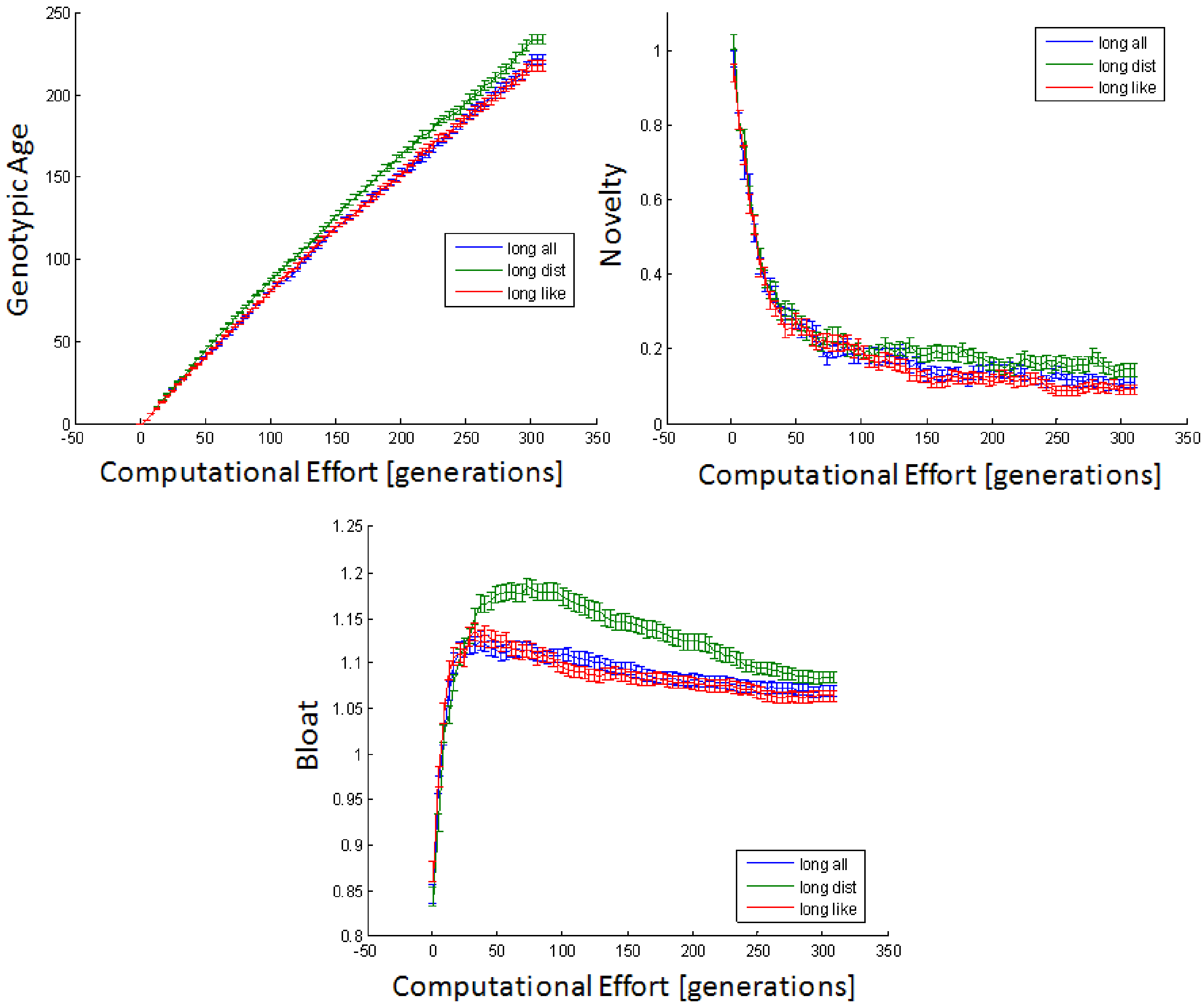

Figure 6. Traits of the best model over time during the evolutionary search. The top left plot shows the genotypic age of the best solution (the number of generations any part of the solution existed in the population). The top right shows the novelty of the best solution (how different it is from the rest of the population). The bottom pane shows the bloat of the best solution (ratio its complexity with the target solution complexity). Error bars indicate the standard error.

metric tended to reach higher bloat, which may be a reflection that it was less likely to converge to the target.

One final observation is that for these traits in Figure 6, there appears to be very little difference between the likelihood metric and the two-component metric. The key difference is only in the overall performance (Figure 4). This suggests that the role of the distance component is to help models move toward the data so that the likelihood component can be used, and does not impact other aspects of the population or evolutionary algorithm.

6. CONCLUSIONS

In this paper we introduced an automated algorithm for identifying stochastic reaction models. The proposed method used an evolutionary algorithm to identify a maximum likelihood set of reactions and reaction coefficients. Instead of only optimizing likelihood, the proposed algorithm used a two-component fitness metric that optimized the distance of a candidate model's distribution from the data point when the likelihood was too small to provide an accurate search gradient.

The experiments indicate that the likelihood metric alone performs well on data with short time gaps in data set. However, when the data set contained large time gaps, where the state of the system evolved far from the local behavior the two-component fitness metric performed best, finding the exact target solution faster and more reliably. Observations on the age, novelty, and bloat of the best solution indicate that the algorithm avoids local optima, and could scale well with increasing complexity systems.

7. ACKNOWLEDGMENTS

This work was supported in part by NIH NIDA grant RC2 DA028981, NSF CDI Grant ECCS 0941561, and DTRA grant HDTRA 1-09-1-0013. The content of this paper is solely the responsibility of the authors and does not necessarily represent the official views of the sponsoring organizations.

8. REFERENCES

[1] D. T. Gillespie, "Exact Stochastic Simulation of Coupled Chemical Reactions," *The Journal of Physical Chemistry,* vol. 81, pp. 2340-2361, 1977.

[2] D. T. Gillespie, "Stochastic simulation of chemical kinetics," *Annu Rev Phys Chem,* vol. 58, pp. 35-55, 2007.

[3] J. L. Doob, "Markoff chains--denumerable case " *Trans. Amer. Math. Soc.,* vol. 58, pp. 455-473, 1945.

[4] Y. Cao, D. T. Gillespie, and L. R. Petzold, "Avoiding negative populations in explicit Poisson tau-leaping," *J Chem Phys,* vol. 123, p. 054104, Aug 1 2005.

[5] S. Hideaki and S. Shigeru, "A Method for Selecting the Bin Size of a Time Histogram," *Neural Comput.,* vol. 19, pp. 1503-1527, 2007.

[6] M. Rosenblatt, "Remarks on some nonparametric estimates of a density function," *Ann. Math. Statist.,* vol. 27, pp. 832-837, 1956.

[7] E. Parzen, "On Estimation of a Probability Density Function and Mode," *The Annals of Mathematical Statistics,* vol. 33, pp. 1065-1076, 1962.

[8] G. Terrell and D. Scott, "Variable kernel density estimation," *The Annals of Statistics,* vol. 20, pp. 1236-1265, 1992.

[9] I. S. Abramson, "On bandwidth variation in kernel estimates--a square root law," *Ann. Statist.,* vol. 10, pp. 1217-1223, 1982.

[10] A. J. Lotka, *Elements of physical biology*. Baltimore: Williams & Wilkins Co., 1925.

[11] V. Volterra, "Variazioni e fluttuazioni del numero d'individui in specie animali conviventi," *Mem. R. Accad. Naz. dei Lincei,* vol. 2, 1926.

[12] G. S. Hornby, "ALPS: the age-layered population structure for reducing the problem of premature convergence," in *GECCO 2006: Proceedings of the 8th annual conference on Genetic and evolutionary computation*. vol. 1, ACM SIGEVO (formerly ISGEC), 2006, pp. 815--822.

[13] J. Lehman and K. O. Stanley, "Abandoning Objectives: Evolution through the Search for Novelty Alone," *Evol Comput,* p. 24, Sep 24 2010.

[14] W. Banzhaf and W. B. Langdon, "Some considerations on the reason for bloat," in *Genetic Programming and Evolvable Machines*. vol. 3, 2002, pp. 81--91.