

АНАЛИЗ МЕТАЭВРИСТИЧЕСКИХ МЕТОДОВ ИДЕНТИФИКАЦИИ ПАРАМЕТРОВ ДЛЯ АГЕНТНЫХ МОДЕЛЕЙ

А.И. Влад

Институт вычислительной математики им. Г.И. Марчука РАН
Россия, 119333, Москва, ул. Губкина, 8
E-mail: firkhraag@gmail.com

Т.Е. Санникова

Институт вычислительной математики им. Г.И. Марчука РАН
Россия, 119333, Москва, ул. Губкина, 8
E-mail: te_san@yahoo.com

Ключевые слова: агентные модели, задача идентификации параметров, метаэвристические алгоритмы.

Аннотация: При моделировании сложных систем с использованием агентного подхода возникает проблема выбора метода идентификации параметров модели. Данная проблема усугубляется тем, что пространство параметров в сложных агентных системах может иметь большую размерность, а время, необходимое для выполнения одного численного эксперимента, довольно большим. Альтернативой традиционным методам оптимизации являются так называемые метаэвристические алгоритмы, дающие приближенное решение за приемлемое время. Целью данного исследования является сравнение различных метаэвристических методов идентификации параметров и анализ их эффективности на примере сложной агентной модели с большой вычислительной сложностью. В качестве алгоритмов рассматривались: алгоритм Метрополиса-Гастингса, метод Монте-Карло с марковской цепью, метод суррогатного моделирования и алгоритм оптимизации рою частиц. Рассматриваемые алгоритмы тестировались на агентной модели динамики семи респираторных вирусных инфекций в Москве.

1. Введение

Агентные модели [1] используются для моделирования систем различной сложности, динамика функционирования которых определяется не глобальными правилами и законами, а индивидуальной деятельностью составляющих ее элементов. Поскольку реальную систему невозможно полностью воспроизвести в виде математической модели, необходимо делать ряд предположений, в частности, какие параметры использовать для описания системы и какие значения им задать для корректной работы модели. Данная проблема усугубляется тем, что пространство параметров модели может иметь большую размерность, что делает задачу отыскания оптимального решения крайне затруднительной в условиях повышенной требовательности агентных моделей к аппаратным и программным мощностям вычислительных систем. Традиционные методы оптимизации не позволяют найти оптимальное решение за разумное время из-за вычислительной сложности агентных моделей. Альтернативным подходом является использование метаэвристических

алгоритмов для решения задачи оптимизации, которые позволяют найти решения, близкие к оптимальным, но с меньшими вычислительными затратами.

В данной работе мы провели анализ различных метаэвристических методов для решения задачи идентификации параметров агентной модели. В качестве методов мы рассматривали: алгоритм Метрополиса-Гастингса, метод Монте-Карло с марковской цепью, метод суррогатного моделирования и алгоритм оптимизации роем частиц, гиперпараметры которых выбирались вручную. В качестве модели была выбрана агентная модель динамики острых респираторных вирусных инфекций (ОРВИ) в мегаполисе с населением около 10 миллионов человек.

2. Агентная модель динамики респираторных вирусных инфекций

Для анализа методов идентификации параметров мы используем агентную модель совместной циркуляции семи типов респираторных вирусов (гриппы типов А и В, риновирусы, респираторно-синцитиальные вирусы, аденовирусы, парагрипп и коронавирусы (НКУ1, NL63, ОС42 и С229Е)) в большом городе [2]. Свойства вирусов, такие как средняя вирусная нагрузка, продолжительности инкубационного периода и периода болезни и вероятность развития симптомов с учетом возрастов носителей, оценивались на основе доступной литературы.

В качестве объекта моделирования был выбран город Москва. Модель состоит из 10 миллионов агентов, представляющих из себя жителей города, которые характеризуются набором свойств. Такие свойства, как возраст, пол и социальный статус, устанавливаются с использованием демографических и социально-экономических данных по различным муниципалитетам города в соответствии с данными Всероссийской переписи 2010 года. Эпидемиологические свойства включают в себя общий уровень иммуноглобулинов, зависящий от возраста, и состояние здоровья агента. Каждый агент может быть восприимчивым или частично восприимчивым к различным вирусам, инфицированным каким-то одним вирусом или резистентным ко всем вирусам, в зависимости от стадии инфекции. Агенты присваиваются домохозяйствам, а также другим коллективам, к которым относятся группы образовательных учреждений и рабочие коллективы. Контакты агентов друг с другом моделируются с использованием сетей контактов, представленных в виде полных графов для домохозяйств и графов Барабаши-Альберт [3] для групп образовательных учреждений и рабочих коллективов с параметрами присоединения, равными 10 и 5, соответственно. Модель воспроизводит один год, начиная с 1 августа, и имеет дискретный шаг по времени, равный одному дню.

Передача вирусов происходит при контактах между восприимчивыми или частично восприимчивыми и инфицированными агентами. Риск инфицирования восприимчивого или частично восприимчивого агента i включает в себя риски заражения каждым моделируемым вирусом v в каждом моделируемом коллективе c , который посещает агент i на шаге модели t , от каждого инфицированного агента, с которым агент i совершает контакт. Риск передачи инфекции определяется произведением пяти независимых рисков: риск, зависящий от скорости выделения вирусных частиц инфицирующим агентом, риск, зависящий от общего уровня иммуноглобулинов инфицируемого агента, риск, зависящий от специфического иммунитета инфицируемого агента, риск, зависящий от продолжительности контакта между агентами и риск, зависящий от температуры воздуха на текущем шаге модели.

В модели имеются 26 настраиваемых параметра, разделенных на 6 групп: группа, отвечающая зависимости риска инфицирования от продолжительности контакта (δ),

группа, отвечающая зависимости риска инфицирования от общего уровня иммуноглобулинов (β_i), группа, отвечающая зависимости риска инфицирования от температуры воздуха (γ_i), группа средних продолжительностей иммунитета к каждому моделируемому вирусу (r_i), где $i = 1, \dots, 7$ – индекс моделируемого вируса, и группа вероятностей инфицирования от неустановленного источника (p_j), где $j = 1, \dots, 4$ – индекс возрастной группы агентов.

Модель используется для воспроизведения выявленной еженедельной заболеваемости ОРВИ в Москве для различных возрастных групп и вирусов за 1997–2002 гг. Они представлены в виде трех групп кривых заболеваемости: общая кривая заболеваемости, кривые заболеваемости для возрастных групп 0–2, 3–6, 7–14 и 15+ лет, а также кривые заболеваемости для семи моделируемых вирусов.

3. Методы идентификации параметров

3.1. Метод Метрополиса-Гастингса

Метод Метрополиса-Гастингса [4] используется для построения вероятностного распределения значений параметров модели.

На первом этапе определяются интервалы возможных значений параметров и задаются их начальные значения. Далее, на каждом шаге i алгоритма генерируется новый набор значений параметров модели $\Theta'_i = \{\theta'_{ij}\}_{j=1}^{26}$ на основе уже имеющегося набора параметров с предыдущего шага Θ_{i-1} :

$$\theta'_{ij} = \frac{\theta_j^{\max} \exp(\delta_{ij}) + \theta_j^{\min}}{1 + \exp(\delta_{ij})}, \delta_{ij} \sim \mathcal{N}\left(\log\left(\frac{\theta_{i-1j} - \theta_j^{\min}}{\theta_j^{\max} - \theta_{i-1j}}\right), \sigma^2\right),$$

где $\theta_{ij} \in [\theta_j^{\min}, \theta_j^{\max}]$, σ^2 – гиперпараметр алгоритма, представляющий из себя дисперсию для генерации новых значений параметров (считаем $\sigma^2 = 0,1$).

Апостериорное распределение вектора параметров Θ_i исследуется с помощью марковской цепи Монте-Карло, применяемой к полученному априорному распределению, которое мы считаем $p(\theta_{ij}) \sim U(\theta_j^{\min}, \theta_j^{\max})$ и правдоподобию числа диагностированных случаев x_{wav} заражения вирусом v для каждой недели w в каждой возрастной группе a в результате численного эксперимента, которые мы считаем независимыми друг от друга и аппроксимированными нормальными распределениями со средними значениями μ_{wav} , полученными из данных, и дисперсией, равной средним значениям (аппроксимация распределения Пуассона).

$$L'_i = \sum_{wav} (-\log(\sqrt{2\pi}\sigma_{wav}) - 0,5 \left(\frac{x_{wav} - \mu_{wav}}{\sigma_{wav}}\right)^2),$$

где $\sigma_{wav} = \sqrt{\mu_{wav}}$. После чего мы либо принимаем новый набор параметров Θ'_i с вероятностью $\alpha(\Theta', \Theta^k)$ ($\Theta_{i+1} = \Theta'_i$), либо отвергаем его ($\Theta_{i+1} = \Theta_i$):

$$(1) \quad \alpha(\Theta'_i, \Theta_i) = \min(1, \exp(L'_i - L_i)).$$

Во избежание попаданий в локальные минимумы мы принимаем новый набор параметров после 10 последовательных отказов. Повторяем данный алгоритм, пока не достигнем заданного числа итераций.

3.2. Метод Монте-Карло с марковской цепью

Данный метод, использованный в работе [2], аналогичен методу Метрополиса-Гастингса (см. главу 2.1), однако в данном случае вместо задания вероятностных распределений мы используем функцию потерь, заданную в виде нормализованной средней абсолютной ошибки:

$$(2) \quad nMAE = \frac{\sum_{wav} |y_{wav}^d - y_{wav}^m|}{\sum_{wav} y_{wav}^d},$$

где y_{wav}^d, y_{wav}^m – диагностированная заболеваемость вирусом v на 1000 человек за неделю w в возрастной группе a по данным и заболеваемость, полученная в результате численного эксперимента из модели, соответственно. Таким образом вероятность $\alpha(\Theta'_i, \Theta_i)$ (1) того, что набор параметров Θ'_i будет принят на очередном шаге алгоритма равна 1 ($\Theta_{i+1} = \Theta'_i$), если $nMAE_i < nMAE_{min}$, и 0 ($\Theta_{i+1} = \Theta_i$) – в противном случае, где $nMAE_i$ – значение функции потерь на шаге i , $nMAE_{min}$ – наименьшее найденное значение функции потерь. В остальном данный метод полностью идентичен методу Метрополиса-Гастингса.

3.3. Метод суррогатной модели

Суррогатный подход к решению задачи идентификации параметров основан на работе [5]. Данный алгоритм характеризуется тем, что результаты, полученные по результатам численных экспериментов с агентной моделью, аппроксимируются с помощью так называемой суррогатной модели. Процесс обучения суррогатной модели, которая должна отражать структуру агентной модели, состоит в следующем:

На начальном этапе проводим 1000 численных экспериментов с использованием наборов параметров, полученных с помощью метода выборки латинского гиперкуба [6] 1000-го порядка с размерностью 26. Латинский гиперкуб N -го порядка k -й размерности и строится путем разбиения интервалов возможных значений каждого из k параметров на N интервалов одинаковой длины, после чего из каждого из этих интервалов случайным образом берется значение, которое потом случайным образом соединяется со значениями других параметров. Одному значению одного параметра может соответствовать только одно значение другого параметра. Таким образом получаем выборку из N наборов значений для k параметров.

Для результатов численного эксперимента вводим функцию потерь, характеризующую отклонение полученных результатов из модели от реально наблюдаемых данных. В данной работе используется нормализованная средняя абсолютная ошибка $nMAE$ (2). Таким образом мы получаем отображение $\Theta_i \rightarrow nMAE_i$ для каждого численного эксперимента $i = 1, \dots, N$.

Для обучения суррогатной модели на полученном тренировочном наборе параметров $X_{i+1} = \cup_{k=1}^i \Theta_k$ с метками $Y_{i+1} = \cup_{k=1}^i nMAE_k$ используется алгоритм XGBoost [7]. Данный алгоритм представляет собой градиентный бустинг, создающий ансамбль решающих деревьев. В качестве целевой функции берется функция среднеквадратичной ошибки. Для данного алгоритма мы задали максимальную глубину обучаемых деревьев решений равной 10, количество раундов для бустинга – 150, скорость обучения – 0,1.

Далее, используем метод Монте-Карло с марковской цепью (см. главу 2.2), но в данном случае без проведения численных экспериментов с моделью, а вместо этого используем обученную суррогатную модель для прогнозирования значения функции потерь $nMAE$. Таким образом получаем набор значений параметров, дающий минимум функции потерь в соответствии с суррогатной моделью, уже для которого проводится численный эксперимент с агентной моделью.

После этого обновляем обучающую выборку $X_{i+1} = X_i \cup \Theta_i$ и набор меток $Y_{i+1} = Y_i \cup nMAE_i$ и повторяем данный алгоритм, пока не достигнем либо удовлетворительного значения функции потерь, либо заданного числа итераций.

3.4. Метод оптимизации роением частиц

Данный метод численной оптимизации имитирует поведение частиц, состояние которых характеризуется положением в пространстве параметров модели, а также вектором скорости перемещения [8]. Кроме того, частицы хранят положение лучшего найденного решения конкретно данной частицей, а также лучшее решение среди всех взятых частиц.

На первом шаге алгоритма задаются число частиц, значения гиперпараметров алгоритма, а также положения частиц в пространстве и начальный вектор скорости. В данной работе мы задаем число частиц, равное 20, а для их начальных положений выбираем 20 лучших найденных решений (минимальные значения функции потерь $nMAE$ (2)) по 1000 численных экспериментов с использованием наборов параметров, полученных с помощью метода выборки латинского гиперкуба (см. пункт 1 главы 2.3). В качестве начального вектора скорости брался нулевой вектор.

Далее, для каждой частицы определяется вектор ее скорости по формуле:

$$v_{ij} = wv_{ij} + c_1r_1(p_{ij} - x_{ij}) + c_2r_2(g_j - x_{ij}),$$

где v_{ij} – j -я компонента скорости частицы i , w , c_1 , c_2 – гиперпараметры алгоритма, задающие вес и ускорения частицы (мы считаем, что $w = c_1 = c_2 = 0,5$), $r_1, r_2 \sim U(0,1)$, x_{ij} – j -я компонента положения частицы i , p_{ij} – j -я компонента лучшего найденного частицей i положения, g_j – j -я компонента лучшего найденного решения всеми частицами.

Обновляем положение частицы: $x_i = x_i + v_i$ и проводим численный эксперимент с полученным набором параметров. Если полученное значение $nMAE < nMAE_i$ или $nMAE < nMAE_{min}$, то обновляем значения p_i и g , соответственно.

Повторяем данный алгоритм, пока не достигнем либо удовлетворительного значения функции потерь, либо заданного числа итераций.

4. Результаты

Результаты сравнения различных методов приведены на рис. 1. Из рисунка видно, что метод Монте-Карло с марковской цепью показал наилучшую сходимость к данным при заданной функции потерь в виде $nMAE$ (2), тогда как другие методы не могли уменьшить значение функции потерь ниже 0,7 за аналогичное время.

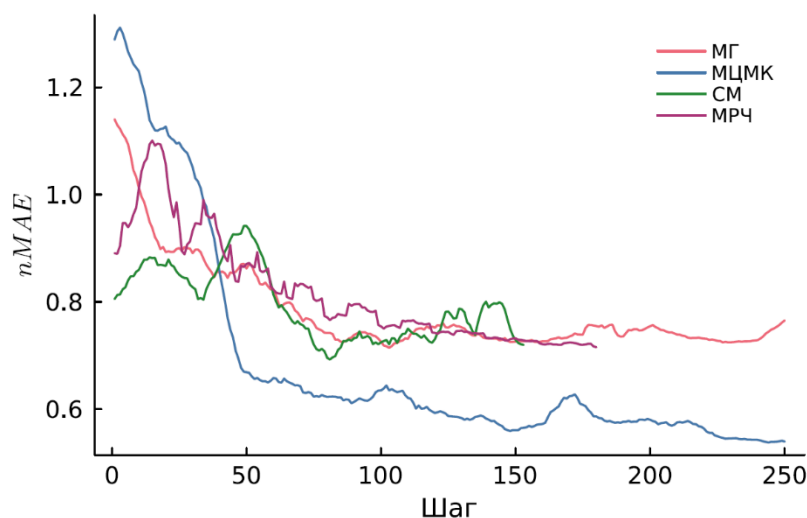


Рис. 1. Сравнение значений функции потерь $nMAE$, полученной на каждом шаге при использовании различных методов для идентификации параметров модели. МГ – метод Метрополиса-Гастингса, МЦМК – метод Монте-Карло с марковской цепью, СМ – метод с использованием суррогатной модели, МРЧ – метод оптимизации роем частиц.

5. Выводы

Метод Монте-Карло с марковской цепью позволяет быстро решать задачу идентификации параметров агентной модели с удовлетворительной точностью. В отличие от подхода с использованием суррогатной модели или метода оптимизации роем частиц, ему не нужно предварительно проводить множество численных экспериментов, а достаточно задать случайное начальное значение, которое может основываться на знаниях о предметной области. Остальные методы оказались более подвержены попаданию в локальные минимумы, что может быть связано с недостаточно хорошим выбором значений гиперпараметров алгоритмов.

Список литературы

1. Macal C.M., North M.J. Tutorial on agent-based modeling and simulation // *Journal of Simulation*. 2010. Vol. 4, No. 3. P. 151-162.
2. Vlad A.I., Romanyukha A.A., Sannikova T.E. Circulation of Respiratory Viruses in the City: Towards an Agent-Based Ecosystem model // *Bulletin of Mathematical Biology*. 2023. Vol. 85, No. 10. P. 100.
3. Albert R., Barabasi A.L. Statistical mechanics of complex networks // *Reviews of modern physics*. 2002. Vol. 74, No. 1. P. 47.
4. Hastings W.K. Monte Carlo Sampling Methods Using Markov Chains and Their Applications // *Biometrika*. 1970. Vol. 57, No. 1. P. 97-109.
5. Lamperti F., Roventini A., Sani A. Agent-based model calibration using machine learning surrogates // *Journal of Economic Dynamics and Control*. 2018. Vol. 90. P. 366-389.
6. McKay M.D., Beckman R.J., Conover W.J. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code // *Technometrics*. 2000. Vol. 42, No. 1. P. 55-61.
7. Chen T., Guestrin C. Xgboost: A scalable tree boosting system // *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016. P. 785-794.
8. Kennedy J., Eberhart R. Particle swarm optimization // *Proceedings of IEEE International Conference on Neural Networks*. 1995. Vol. 4. P. 1942-1948.