Finding the maximum-a-posteriori expressed behaviour of agents in an agent-based model *** UNFINISHED DRAFT ***

Daniel Tang
Leeds Institute for Data Analytics
University of Leeds
Leeds, UK
D.Tang@leeds.ac.uk

February 22, 2020

Abstract

We often make partial, noisy observations of some dynamical system and wish to explain those observations in terms of the most probable set of events that caused them. Here we consider the case when the dynamical system is a (possibly stochastic) time-stepping agent-based model with a discrete state space, the (possibly noisy) observations are the number of agents that have some given property and the events are the expressed behaviours of the agents.

We show that this problem can be reduced to an integer linear programming problem which can subsequently be solved numerically using a standard branch-and-cut algorithm. We describe two implementations, an "offline" algorithm that finds the maximum-a-posteriori expressed behaviours given a set of observations over a finite time window, and an "online" algorithm that incrementally builds a feasible set of behaviours from a stream of observations that may have no natural beginning or end.

We demonstrate both algorithms on a spatial predator-prey model on a 32x32 grid with an initial population of 100 agents.

This is an unfinished draft which may contain errors and is subject to change.

Keywords Data assimilation, Agent based model, Quantum field theory, Probabilistic programming

1 Introduction

Intuitively, the problem we'll consider here is as follows: there is a complex-system of interest which we believe can be well modelled by an agent based model. We have some prior beliefs about the boundary conditions of the system and have made some observations over some period of time. However, the observations are incomplete (in that they do not specify the full state of the model at any time) and possibly noisy. The problem is to find the maximum-a-posteriori (MAP) behaviour expressed by each agent given the observations. The expressed behaviours should consist of a feasible set of agent actions or *events* that lead up to, or *explain*, the observations.

In order to make an algorithm to solve this problem, we'll first define the problem a bit more precisely.

1.1 What is an agent-based model?

We take the essential characteristic of an agent-based model to be that the dynamics of the model can be described entirely in terms of agent behaviours (if the environment contains objects, these are also modelled as agents). Each agent may have an internal state and the state of the whole model at any instant can be completely specified by the states of all the agents.

Within this broad definition, there are a number of sub-categorizations

- Modelled time can be either discrete or continuous. In a discrete-time model, agent behaviour consists of timesteps from time t to $t + 1$ and time is an integer. In a continuous-time model agent behaviour consists of discrete events that can happen at any time expressed as a real number. In this paper we will consider discrete-time models.

- The internal state of an agent can be discrete, continuous or mixed depending on whether the state can be most naturally expressed as a vector of integers, real numbers or a mixture of both. In this paper we will consider agents with discrete-state.
- Agents can be deterministic or stochastic. A deterministic agent will always exhibit the same behaviour if its own internal state and the states of other agents in the model are fixed, whereas a stochastic agent will exhibit different behaviours randomly drawn from a fixed distribution under the same circumstances. In this paper we will consider stochastic agents.
- If only one agent can occupy a given state at any instant, then the state of the model can be described as a set of agent states and we call it a Fermi-Dirac model, whereas if two or more agents can occupy the same state at the same time then the model state must be described as a multiset(Blizard et al., 1988) of agent states and we call it a Bose-Einstein model. In this paper we will consider Bose-Einstein models.

We define the behaviour of an agent as a computer program equipped with the ability to

- query and modify its own state
- query whether there exist any agents in the model that satisfy any given predicate on their state
- create a new agent in any state
- remove itself from the model
- call a function that returns a random number.

Such a program can be expressed as a conditional probability distribution over a set of events (Staton, Wood, Yang, Heunen, & Kammar, 2016). Each event represents an agent performing an action that expresses a specific behaviour, and has the form $(\Phi|\psi, R, \bar{R})$ where ψ is the internal state of the agent before the behaviour is expressed, R is the set of other agents (states) that must be present for the behaviour to be expressed, $\bar{R}$ is the set of agents that must be absent for the behaviour to be expressed and Φ is the multiset of agents that are the result of the action occurring (including the final state of the acting agent). So, if we say an agent has a behaviour $P(\Phi|\psi, R, \bar{R})$ then when it is in state ψ in an environment containing all agents in R and no agents in $\bar{R}$, it performs an action that results in Φ with probability $P(\Phi|\psi, R, \bar{R})$.

So, for example, an agent with behaviour $P(\{\psi'\}|\psi, \emptyset, \emptyset) = 1.0$ will, with certainty, change its state to ψ' at time $t + 1$ if it is in state ψ at time t. An agent with behaviour $P(\emptyset|\psi, \{\phi\}, \emptyset) = 0.5$ in state ψ in the presence of another agent in state ϕ will disappear with probability 0.5.

Without loss of generality, we assume that all agents in the model have the same behaviour. If necessary, multiple types of agent can be modelled by making the agent's type part of its state.

In any timestep of the model a multiset of events will occur, we will call such a multiset a *model event* to distinguish it from the *agent events* that are members of the model event.

We define an event trajectory as a list of model events $\langle E_1...E_n\rangle$, which can be thought of as the assertion that, for each i, the events in E_i occurred in timestep from $t = i - 1$ to $t = i$.

If a model is in state Ψ, only certain model events are possible so we say that Ψ *satisfies* a model event E, and write $\Psi \vdash E$, if the following conditions hold:

Each agent in Ψ and must perform exactly one action per timestep

$$\sum_{(\Phi|\psi,R,\bar{R})\in E} \{\psi\} = \Psi \tag{1}$$

...each event's presence-requirements must be fulfilled

$$\bigcup_{(\Phi|\psi,R,\bar{R})\in E} R \subseteq \Psi$$

...and each event's absence-requirements must be fulfilled

$$\left(\bigcup_{(\Phi|\psi,R,\bar{R})\in E_t} \bar{R} \right) \cap \Psi = \emptyset$$

The consequence of a model event is the sum of the consequences of its members. We'll write this as $\Psi(E)$

$$\Psi(E) = \sum_{(\Phi|\psi,R,\bar{R})\in E} \Phi \tag{2}$$

The consequences of an event trajectory, $\langle E_1...E_n\rangle$, is a trajectory of states $\langle\Psi(E_1)...\Psi(E_2)\rangle$ and the trajectory is feasible if

$$\forall i \in 2...n : \Psi(E_{i-1}) \vdash E_i$$

The probability of a feasible model event is given by

$$P(E|\Psi) = \prod_{(\Phi|\psi,R,\bar{R})\in E} P(\Phi|\psi, R, \bar{R})$$

So, if the model is in state Ψ_t at time t then the set of possible timesteps is the set of model events that are satisfied by Ψ, which will each occur with probability $P(E|\Psi)$. At time $t+1$, after the occurence of model event E, the state will be $\Psi_{t+1} = \Psi(E)$.

1.2 What is an observation?

In this paper we will consider observations that consist of a count of the number of agents for which some predicate is true. The observation can be exact or noisy in that the real count can lie in some range. More formally $\omega = \langle L, U, B\rangle$ denotes the observation that the number of agents for which predicate B is true is somewhere in the range $L \leq N \leq U$. If Ψ is a model state, then we say that Ψ satisfies the observation, which we write as $\Psi \vdash \omega_i$, if and only if

$$L \leq |\{\psi : \psi \in \Psi \wedge B(\psi)\}| \leq U$$

From this, we say that Ψ satisfies a set of observations, Ω, if it satisfies all of its members, and a trajectory of states $\mathcal{T} = \langle\Psi_1...\Psi_n\rangle$ satisfies a list of sets of observations (which we'll call an observation of a trajectory), $\mathcal{O} = \langle\Omega_1...\Omega_n\rangle$ if for each i, Ψ_i satisfies Ω_i.

This form of observation can deal with exact observations ($L = U$), observations where there may be false-positives (U is the observed count and $L = 0$), observations where there may be false negatives (L is the observed count and $U = \infty$) or observations that have some uncertainty ($L = N - \Delta N$ and $U = N + \Delta N$).

1.3 What is a prior belief about boundary conditions?

The boundary conditions on the system consist of agents entering the system at particular times. At its simplest, it is a set of agents that enter the system at time $t = 0$, though they need not all enter at this time.

More generally, it is a set of probability distributions $P(\Gamma, t)$ which give the prior probability that a multiset Γ is a subset of the agents that enter the system at time t.

In this paper we assume that these prior beliefs can be factorised into factors that involve single agent states. In this case the prior probabilities can be expressed as

$$P(\Gamma, t) = \prod_{\psi \in^{=n} \Gamma} P(\psi, n, t)$$

where $\psi \in^{=n} \Gamma$ is true when ψ appears in Γ exactly n times and $P(\psi, n, t)$ is the probability of at least n agents in state ψ being injected at time t.

Boundary conditions can be conveniently represented in terms of *injection events* which are caused by a set of hypothetical (though never explicitly represented) *zero energy* agents $\emptyset_\psi$ which, by definition, are present before $t = 0$. An injection event , $(\{\emptyset_\psi, \psi^n\}|\emptyset_\psi, \emptyset, \emptyset)$, has no requirements and results in n agents in the state ψ being injected into the system, along with the continued existence of the injecting agent. If we allow the behaviour of these agents to be time dependent, then any set of injection events can be represented by supposing the existence of zero energy agent at time $t = -1$ for each state that needs injection.

With zero energy agents we can now conveniently define an event trajectory along with its boundary conditions as a list of model events $\mathcal{E} = \langle E_0...E_n\rangle$ which induces a model state trajectory $\mathcal{T} = \langle\Psi(E_0)...\Psi(E_n)\rangle$ and which is feasible if and only if $\{\emptyset_{\psi_1}...\emptyset_{\psi_n}\} \vdash E_0$ and $\forall i \in [1, n] : \Psi(E_{i-1}) \vdash E_i$

1.4 What is an explanation of a set of observations?

Armed with the above definitions, we can define an "explanation" of an observation of a trajectory, $\mathcal{O} = \langle\Omega_0...\Omega_n\rangle$, as a feasible event trajectory $\mathcal{E} = \langle E_0...E_n\rangle$ whose associated state trajectory $\mathcal{T} = \langle\Psi(E_0)...\Psi(E_n)\rangle$ satisfies the observation $\mathcal{O}$.

The probability of an explanation is just the product of the probabilities of all the events

$$P(\langle E_0...E_m\rangle) = \prod_i \prod_{\epsilon\in E_i,} P(\epsilon)$$

2 Finding the MAP event trajectory

Bringing all this together, we can now express our problem precisely:

Given a set of observations Ω and an agent behaviour $P(\Phi|\psi, R, \bar{R})$, find

$$\mathcal{E} = \arg \max_{\langle E_0 \dots E_m \rangle} \prod_i \prod_{\epsilon \in E_i,} P(\epsilon) \tag{3}$$

subject to $\mathcal{E}$ being feasible and it's implied state trajectory satisfying Ω.

To turn this into a tractable algorithm, first represent an event trajectory as a set of non-negative integers c_{te} such that in timestep t, the event e occurs exactly c_{te} times.

Now encode each event in an agent's behaviour, $e = (\Phi|\psi, R, \bar{R})$, by letting

$$\Phi_{e\phi} = n : \phi \in^n \Phi \tag{4}$$

$$\psi_{e\phi} = \begin{cases} 1 & \text{if } \phi = \psi \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

$$R_{e\phi} = \begin{cases} 1 & \text{if } \phi \in R \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

$$\bar{R}_{e\phi} = \begin{cases} 1 & \text{if } \phi \in \bar{R} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

Now encode the observation predicates for each event's consequences

$$B_e = |\{\psi : \psi \in \Phi \wedge B(\psi)\}| \tag{8}$$

and encode the probability of each event

$$P_e = P(\Phi|\psi, R, \bar{R})$$

finally, let

$$\Psi_{t\phi} = \sum_e \Phi_{e\phi} c_{te} \tag{9}$$

and define a set of boolean variables $0 \leq b_{t\phi} \leq 1$ which satisfy

$$0 \leq M b_{t\phi} - \Psi_{t\phi} \tag{10}$$

$$0 \leq \Psi_{t\phi} - b_{t\phi} \tag{11}$$

for some *multiplicity* M which can be any number larger than the expected maximum number of agents in the model at any one time.

We can now express the problem as:

Maximise

$$P = \sum_t \sum_e \log(P_e) c_{te} \tag{12}$$

subject to the observations being satisfied

$$\forall \Omega_t \in \mathcal{O}, \langle L, U, B \rangle \in \Omega_t : L \leq \sum_e B_e c_{te} \leq U \tag{13}$$

...each agent performing exactly one action

$$\forall \phi : 0 \leq \Psi_{(t-1)\phi} - \sum_e \psi_{e\phi} c_{te} \leq 0 \tag{14}$$

...each event's presence requirements being fulfilled

$$\forall \phi : 0 \leq M b_{(t-1)\phi} - \sum_e R_{e\phi} c_{te} \tag{15}$$

...each event's absence requirements being fulfilled

$$\forall \phi : \sum_e \bar{R}_{e\phi} c_{te} + M b_{(t-1)\phi} \leq M \tag{16}$$

This can be seen to be in the form of a pure integer linear programming problem, which in full generality is NP complete, but for which there exists many algorithms that, in practice, are able to find solutions to reasonably large problems.

3 Online assimilation

Observations often arrive in a constant stream which may have no specific start or end. In this case, we would usually split the stream up into time *windows* and update the forecasts one window at a time. However, if we commit to the maximum-a-posteriori explanation of one window, and use the end state of this explanation as the start state of the next window, we are in danger of being led down "garden paths"; i.e. the observations lead us into believing the most probable explanation, only to later show us that a much less probable and completely incompatible explanation is in-fact the case. If the garden path is contained within a single window then the optimisation will deal with it, but if it stretches across windows, we may commit to believing events which, in the light of more evidence, turn out to be impossible. At this point the assimilation must stop because there is no event trajectory that satisfies the observations for the current window. To solve this problem, we present a second algorithm for use with streaming observations.

We begin by defining the idea of a *partial event trajectory* which is like a normal event trajectory except that instead of each agent having to perform exactly one action per timestep, they can perform at most one action per timestep. So, in the definition of satisfaction, in place of equation 1 we have

$$\sum_{(\Phi|\psi,R,\bar{R})\in E} \{\psi\} \subseteq \Psi \tag{17}$$

In this case, we'll say that Ψ partially satisfies E. Intuitively, a partial event trajectory can be interpreted as saying that at least these events happened in these timesteps, but other additional events may have happened concomitantly.

A solution to the resulting problem is a partial event trajectory that makes as few commitments as possible while satisfying the observations. In a partial trajectory, all observations are satisfied but only a subset of the agents present at any particular time will be moved forward by the next model event, so agents may be left hanging at a time in the middle of the trajectory. This is good, because it reduces the likelihood that we will be led down a garden path by delaying any commitment to the exhibited behaviour of the hanging agents until more observations are available.

Once we've found a partial trajectory, we take the next window of observations and find the maximum-a-posteriori partial trajectory given these new observations and the partial trajectory we've already committed to. In order to correctly move hanging agents forward in time, we need to distinguish between the events that we've already committed to and the events we're optimising over. If we let $\hat{c}_{te}$ be the number of events of type e that we've committed to in timestep t, then equation 18 becomes

$$\Psi_{t\phi} - \sum_e \Phi_{e\phi} c_{te} = \sum_e \Phi_{e\phi} \hat{c}_{te} \tag{18}$$

the agency constraint, equation 14, becomes

$$\forall \phi : \sum_e \psi_{e\phi} \hat{c}_{te} \leq \Psi_{(t-1)\phi} - \sum_e \psi_{e\phi} c_{te} \tag{19}$$

and the absence constraint, equation 16, becomes

$$\forall \phi : \sum_e \bar{R}_{e\phi} c_{te} + M b_{(t-1)\phi} \leq M - \sum_e \bar{R}_{e\phi} \hat{c}_{te} \tag{20}$$

Since observations are fully satisfied by a partial trajectory, the observation constraints, equation 13, now only apply to timesteps for which there are no committed events. The presence requirement constraints can remain unchanged.

Even though partial trajectories reduce the probability of us being led down the garden path, they do not eradicate it completely. So, if we end up with observations that are incompatible with our commitments, we rollback the commitments, one timestep at a time, from the most recent until a solution becomes possible. In the worst case, we'll have to roll-back all the way to $t = 0$, but in practice the probability of this will be very small for most models.

One final issue with online assimilation is dying agents (or agents that leave the model for some other reason). When an agent dies it will not be observed again, but the online assimilation will just leave it in the hanging state on the off-chance that it's still around but has evaded observation. At some point we are likely to want to commit to saying that the agent has died. To do this we can estimate the probability of the agent evading observation for so long, and when this estimate falls below a certain level, we commit to it's death. Since all agent events have a probability of at most 1.0, it is most probable that the agent died immediately after last being observed.

4 Experiments with a spatial predator-prey model

In order to demonstrate the above algorithms, we simulated a spatial predator-prey model on a 32x32 grid with periodic boundary conditions. Agents can be either predator or prey. At each timestep, agents can stay on the same grid-square, move to an adjacent grid-square (i.e. up, down, left or right) or die. Prey can also give birth to another prey on an

Aget type	Description	Probability
Prey	die	0.03
	reproduce	0.06
	move (up/down/left/right) / be eaten	0.728
	stay put / be eaten	0.182
Predator	die	0.05
	move (up/down/left/right)	0.76
	stay put	0.19

Table 1: The rates of each behaviour in the predator-prey model

adjacent grid-square. Prey cannot move onto a square that contains a predator, but in the presence of a predator on an adjacent square they may be eaten and the predator reproduce into the newly vacant square. The probability of each behaviour is shown in table 1. Perhaps surprisingly, even a model with such a simple set of behaviours exhibits quite complex emergent properties that are difficult to predict.

A forward simulation of the ABM was performed in order to create observation data and boundary conditions. The boundary conditions consisted of a fully observed initial state while in subsequent timesteps each agent was observed with a probability 0.5.

4.1 Offline MAP experiment

Using the boundary conditions and observations for five timesteps, we set up an integer linear programming problem as specified above and used the Google's OR-Tools(Google, 2020) to solve the problem and find the maximum-a-posteriori trajectory. A capture of a typical configuration of the real state and the MAP at the end of the five timesteps is shown in figure ... Figure ... shows the average over xxx simulations of the Manhattan distance between the unobserved agents in the MAP and the nearest agent of the same type in the real state. For comparison, we also show the average distance for a model that places the agents randomly with uniform probability.

4.2 Online assimilation experiment

5 Conclusion

We have demonstrated that the maximum-a-posteriori trajectory of an agent based model can be found for a window of partial, noisy observaations of a spatial predator-prey model. We have also shown how to construct a feasible interpretation of a stream of such observations of arbitrary length.

Since integer linear programming problems are, in full generality, NP-complete, the technique described here will not scale to very large models but will work best when agents choose from one of a limited number of options at each timestep, when interactions are sparse and when the observations impose strong constraints on the state.

Finding the MAP trajectory can be useful for a number of reasons. If we're forced to commit to a specific trajectory, given some observations, the MAP is our best bet. However, it is the most probable trajectory (perhaps among others with the same probability) among a very large number of other less probable trajectories. So usually it is very unlikely that the MAP is what really happened and we certainly shouldn't, in general, use the MAP as an interpretation of what actually happened. However, the MAP trajectory can be useful in other ways; for example, it is often a good initial sample to start an MCMC sampling algorithm, or showing the non-existence of a MAP can show that there must be either a false observation or some process has occurred that

References

Blizard, W. D., et al. (1988). Multiset theory. *Notre Dame Journal of formal logic*, *30*(1), 36–66.

Google. (2020). *Or-tools.* Retrieved from `https://developers.google.com/optimization` ([Online; accessed 18-February-2020])

Staton, S., Wood, F., Yang, H., Heunen, C., & Kammar, O. (2016). Semantics for probabilistic programming: higher-order functions, continuous distributions, and soft constraints. In *2016 31st annual acm/ieee symposium on logic in computer science (lics)* (pp. 1–10).