# Dynamics of innovation along foraged trajectories

S Ganga Prasath

## 1 2D grid world

Let us say an agent has an intrinsic behavior to follow a herd or a trail laid by other ants or a flock of birds. All the agents simply following this herd following behavior will not result in new solutions. Agents often have to innovate and find new solutions either because the trails are not available anymore or because you know a better route (influence of history) or perturbations (intrinsic or environmental) can throw you off trails and you have to find new solutions. We are interested in understanding the role of intrinsic behavior on the dynamics of innovation.

We will start by implementing an agent trying to optimally traverse from point A to point B in a 2D grid world. The steps involved in traditional reinforcement learning based on SARSA algorithm involves essentially 2-steps: $(i)$ action-value function update, $(ii)$ policy update. The action-value update equation in on-policy SARSA is given by

$$\begin{aligned} Q_\pi(s_t, a_t) \leftarrow\ & Q_\pi(s_t, a_t) \\ & + \alpha\{r_{t+1} + \gamma Q_\pi(s_{t+1}, a_{t+1}) - Q_\pi(s_t, a_t)\} \end{aligned} \tag{1}$$

while the policy update is given by

$$\pi(s) = \arg\max_a Q(s, a).$$

This is shown in algorithmic form in Alg. 1.

**Algorithm 1: SARSA algorithm for action-value update**

```
Initialize: State, $s_0$; action, $a_0$; action-value function, $Q(s_j, a_j)$
foreach epoch do
    Update action-value function using SARSA rule:
        $Q_\pi(s_t, a_t) \leftarrow Q_\pi(s_t, a_t) + \alpha\{r_{t+1} + \gamma Q_\pi(s_{t+1}, a_{t+1}) - Q_\pi(s_t, a_t)\}$
    Choose action:
        $\pi(s_t) = \arg\max_a Q(s_t, a_t)$
    Calculate reward, $r_t$
    Check if target $s^*$ is reached, continue if not
end
```

**Algorithm 1:**

# 2 Trail following behavior

Consider a trail of pheromone $\mathbf{x}^*(s)$ represented by arc-length parameterization $s$. The concentration field in 2D is then given by $c(\mathbf{x}) = c_o\delta(\mathbf{x} - \mathbf{x}^*(s))$. This filed of course is assumed to be steady but can be made time-dependent by simply adding a decay time-scale $\tau$ to get: $c(\mathbf{x}, t) = c_o\delta(\mathbf{x} - \mathbf{x}^*(s))\mathrm{e}^{-t/\tau}$. We can immediately evaluate some of the properties of the curve $\mathbf{x}^*(s)$ which will come in handy soon: $\hat{\mathbf{t}}(s) = \mathrm{d}\mathbf{x}^*(s)/\mathrm{d}s = \{\cos\psi(s), \sin\psi(s)\}$ and $\hat{\mathbf{n}}(s) = \{\sin\psi(s), \cos\psi(s)\}$. As we can see the entire curve $\mathbf{x}^*(s)$ can be represented only using $\psi(s)$ up to global translations and rotations, which is a well known property of curves in 2D.

The state of the agent/ant in our problem is represented by its coordinates $\mathbf{r}(t) = \{r_x(t), r_y(t)\}$ and it can make measurements about how far from the pheromone trail it is $\mathrm{d}\mathbf{r}(t)$ as well as the orientation of the trail, $\hat{\mathbf{t}}(s, t)$. The action that it takes from these measurements is to align its orientation, $\hat{\mathbf{p}}(t)$ along a direction that will take it towards the trail. We show in Fig. 2 a schematic of the problem set up.The strategy used by the agent for tracking the pheromone trail will be to move along the direction given by $(\mathrm{d}\mathbf{r}/|\mathrm{d}\mathbf{r}| + \hat{\mathbf{t}})$ by a fixed length $h$. In order to identify the location along $\mathbf{x}^*(s)$ where $\mathrm{d}\mathbf{r}$ intersects, we use the condition $\mathrm{d}\mathbf{r}(t) \perp \hat{\mathbf{t}}(s, t)$.

## 2.1 Semi-circular trail

For the problem at hand, we will considering a semi-circular trail whose coordinates can be written in arc-length form as $\mathbf{x}^*(s) = \{a\cos(s/a), a\sin(s/a)\}$ where $a$ is the radius of the circle. The trail starts at $s = 0$ given by coordinates $\mathbf{x}^*(0) = \{a, 0\}$ and ends at $s = \pi a$

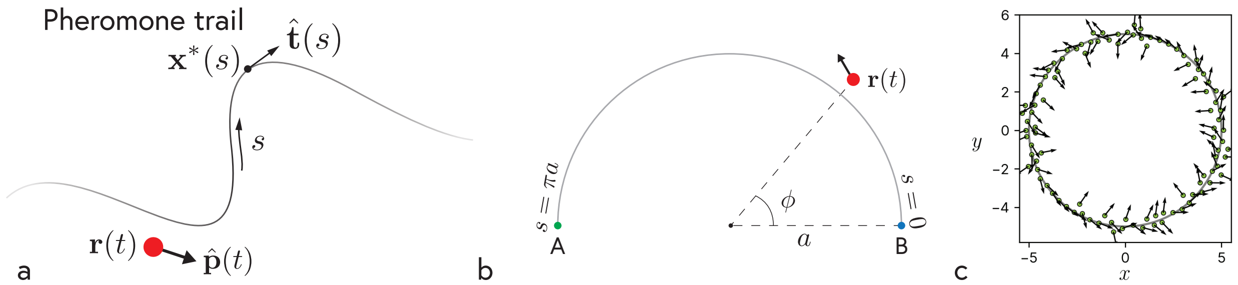

Figure 1: Schematic of setup

at $\mathbf{x}^*(\pi a) = \{-a, 0\}$. From this it is easy to find that $\hat{\mathbf{t}}(s) = \{-\sin(s/a), \cos(s/a)\} = \{\cos(\pi/2 + s/a), \sin(\pi/2 + s/a)\}$, and $\hat{\mathbf{n}}(s) = \{-\cos(s/a), \sin(s/a)\}$. We see that we can represent $\hat{\mathbf{t}}(s)$ through $\psi(s) = (\pi/2 + s/a)$. We can now calculate the location along the arc-length where $\mathbf{r}(t)$ is closest by using the constraint condition $\mathrm{d}\mathbf{r}(t) \perp \hat{\mathbf{t}}(s,t)$ or equivalently $\mathrm{d}\mathbf{r}(t) \cdot \hat{\mathbf{t}}(s,t) = 0$. We can now write $\mathrm{d}\mathbf{r}(t) \sim \{a\cos(s/a) - r_x, a\sin(s/a) - r_y\}$ (up to normalization) and get the location $s_*(t) = a\tan^{-1}(r_y/r_x)$ by using the formula for $\hat{\mathbf{t}}(s)$. From this it is trivial to see that the angle along $\mathrm{d}\mathbf{r}$ is $\phi(t) = \tan^{-1}(r_y/r_x)$ (as is evident in Fig. 2($b$)). For a given location of the agent, $\mathbf{r}(t)$ the orientation it needs to take in the next step can be easily calculated to be $\theta(t) = (\pi/4 - \phi(t))$.

We can now set up the entire dynamics of the agent's trail following behavior on this semi-circle. We can state this in the notation of reinforcement learning as it will become helpful later. The state of the agent is $S^t = \{r_x^t, r_y^t\}$, the measurements it makes are $M^t = \phi^t$ and using this information the action the agent takes is $A^t = \{\theta^t\}$ which is its orientation. The dynamics of the agent can now be written as follows:

$$\text{Measurement update: } \phi^{t+1} = \tan^{-1}\left(\frac{r_y^t}{r_x^t}\right) + \zeta^t, \tag{2}$$

$$\text{Action update: } \hat{\mathbf{p}}^{t+1} = \hat{\mathbf{t}}^{t+1} + \mathrm{d}\mathbf{r}^{t+1}, \tag{3}$$

$$\text{State update: } \mathbf{r}^{t+1} = \mathbf{r}^t + l\hat{\mathbf{p}}^{t+1}, \tag{4}$$

where $\mathbf{r}^t = \{r_x^t, r_y^t\}$, $\mathrm{d}\mathbf{r}^t = (\mathbf{x}^*(s^t) - \mathbf{r}^t)/||\mathbf{x}^*(s^t) - \mathbf{r}(t)||$ and $\hat{\mathbf{p}}^t = \{\cos\theta^t, \sin\theta^t\}$. We have added sensory noise $\zeta^t$ (which is sampled from a uniform distribution) to the measurement to reflect the error that accompanies measurements usually. The solution dynamics is shown in Fig. 2($c$). We call this line-following behavior as a policy $\pi_{\text{ite}}(\mathbf{r}, \phi)$ : which denotes the agent's innate behavior to follow pheromone trails.

# 3 Ornstein-Uhlenbeck process

Let us consider an agent whose orientation $\hat{\mathbf{p}} = (\cos\theta, \sin\theta)$ wants to relax to a neutral orientation with stochasticity associated with estimation error. The dynamics of the orientation can be written as a stochastic differential equation given by

$$\dot{\theta}(t) = -\alpha\theta(t) + \sqrt{2D}\eta(t), \tag{5}$$
$$\langle\eta(t)\eta(0)\rangle = \delta(t). \tag{6}$$

This is the evolution of the famous Ornstein-Uhlenbeck process which has been extremely well studied and the evolution of the probability density can be solved exactly. The Fokker-Plack equation for its dynamics which provides the probability of a given state $\theta(t)$ i.e. $p(\theta, t)$, is given by

$$\frac{\partial p}{\partial t} = \alpha\frac{\partial(\theta p)}{\partial\theta} + D\frac{\partial^2 p}{\partial\theta^2}$$

The dynamics of the angle relaxes to equilibrium angle $\theta = 0$ in the absence of fluctuations. The evolution of the mean is $\dot{h}(t) = -\alpha h(t)$ where $h(t) = \langle\theta(t)\rangle$ while the co-variance dynamics is given by $\dot{\zeta}(t) = -2\alpha\zeta(t) + 2D$ with $\zeta = \langle(\theta(t) - h(t))^2\rangle = \langle\theta^2(t)\rangle - h(t)^2$.

The position of the agent evolves as $\dot{\mathbf{r}}(t) = v_o\hat{\mathbf{p}}(t)$. Under the assumption that the angle is small, we can write $\dot{\mathbf{r}}(t) = v_o(1, \theta)$. Thus we have $\dot{y}(t) = v_o\theta$ which can be used to rewrite the orientation equation as

$$\ddot{y}(t) = -\alpha\dot{y}(t) + v_o\sqrt{2D}\eta(t). \tag{7}$$

This second order equation needs to boundary condition. We require $y(0) = 0$ and $\dot{y}(0) = 0$ which comes from setting the initial orientation to be 0. We are interested in the evolution of the mean and the variance of this SDE. Let $m(t) = \langle y(t)\rangle$ where the averaging is the ensemble average (equivalently the average of the several trajectories of the same dynamics). The mean $m(t)$ evolves as

$$\ddot{m}(t) = -\alpha\dot{m}(t). \tag{8}$$

By applying the above boundary condition, we can immediately see that $m(t) = 0$. Nevertheless, we are interested in the evolution of the standard deviation, $\beta(t) = \langle[y(t) - m(t)]^2\rangle$. In order to derive an expression for that, we differentiate the definition once to get

$$\begin{aligned}\dot{\beta}(t) &= 2\langle\{y(t) - m(t)\}\{\dot{y}(t) - \dot{m}(t)\}\rangle, \\ &= 2\langle y\dot{y}\rangle - 2m\dot{m}, \\ &= 2v_o[\langle y\theta\rangle - m\dot{m}].\end{aligned}$$

We can estimate $\langle y\theta\rangle$ as follows

$$\begin{aligned}\langle y\theta\rangle &= v_o\mathbb{E}\left[\int_0^t \theta(s)\theta(t)\ \mathrm{d}s\right], \\ &= v_o\int_0^t \langle\theta(t)\theta(s)\rangle\ \mathrm{d}s.\end{aligned}$$

We know that for an OU process,

$$\langle \theta(t)\theta(s)\rangle = \frac{D}{\alpha}\Bigg[\mathrm{e}^{-\alpha(t-s)} - \mathrm{e}^{-\alpha(t+s)}\Bigg].$$

From this we see that

$$\begin{aligned}\langle y\theta\rangle &= \frac{v_o D}{\alpha}\int_0^t \Bigg[\mathrm{e}^{-\alpha(t-s)} - \mathrm{e}^{-\alpha(t+s)}\Bigg]\,\mathrm{d}s,\\ &= \frac{v_o D}{\alpha^2}\Bigg[1-2\mathrm{e}^{-\alpha t}+\mathrm{e}^{-2\alpha t}\Bigg].\end{aligned}$$

We thus have the evolution of the co-variance of $y(t)$ as

$$\dot{\beta}(t) = \frac{2v_o^2 D}{\alpha^2}\Bigg[1-2\mathrm{e}^{-\alpha t}+\mathrm{e}^{-2\alpha t}\Bigg]. \tag{9}$$

We finally have the variance as a function of time as

$$\beta(t) = \frac{v_o^2 D}{\alpha^3}\Bigg[2\alpha t-\mathrm{e}^{-2\alpha t}+4\mathrm{e}^{-\alpha t}-3\Bigg]. \tag{10}$$

For $t \gg \alpha^{-1}$ we can immediately see that $\beta(t) \approx 2v_o^2 Dt/\alpha^2$. We are interested in the variance at time, $t^* = c/v_o$ where $c$ is the distance of the agent from the goal A and $v_o$ is the intrinsic speed of the agent. At this time the location of the agent along x-axis is $x(t^*) = c$ while the variance in $y(t)$ is $\beta(t^*) \approx 2v_o Dc/\alpha^2$.

# 4 Evaluating value function of policies

We are at a stage where we can solve the speed/distance-accuracy trade-off for the an agent trying to make optimal decision on when to initiate innovation. In order to do that, let us define clearly what is the state and action of the agent.

## 4.1 $v(s^t)$ for $\pi_{\text{int}}(s^t)$

The agent's state $s^t$ is given by its position, $s^t : \mathbf{r}^t = \{x^t, y^t\}$ and the action it can take at these location is choose a orientation to move, $a^t : \theta^t$. Once the agent choose this action, it is taken to the next state $s^t$ through the dynamics: $\mathbf{r}^{t+1} = \mathbf{r}^t + l\hat{\mathbf{p}}_\theta^t$. Before we compute the state-value function for the intrinsic policy, $\pi_{\text{int}}(s)$, the sequence of process we described above can be written as

$$\text{Measurement: } \phi^t = \tan^{-1}\left(\frac{r_y^t}{r_x^t}\right), \tag{11}$$

$$\text{Action, } a^t\text{: } \theta^t = \frac{\pi}{2} + \phi^t + \zeta^t, \tag{12}$$

$$\text{State update, } s^{t+1}\text{: } \mathbf{r}^{t+1} = \mathbf{r}^t + l\hat{\mathbf{p}}_\theta^t. \tag{13}$$

It is however difficult to evaluate the value function for this specific functional form of the policy. We assume that the distribution of the orientation of the agent is distributed as through it is a worm-like chain polymer whose orientation distribution is given by

$$\mathcal{P}[\theta(s)] = \frac{\mathrm{e}^{-\beta E_b}}{\mathcal{Z}}$$

where the bending energy $E_b[\theta(s)] = B/2 \int_0^L (\theta'(s) - \kappa_o)^2 \, \mathrm{d}s$, fugacity $\mathcal{Z} = \int \mathcal{D}[\theta(s)] \exp\{-\beta E_b[\theta(s)]\}$ and the bending stiffness $B$ captures the constraint for the agent to move forward, $\beta$ is the effective temperature capturing the fluctuations in the trajectory of agent. We also have boundary conditions $\theta(0) = 0$ and $\theta'(0) = \kappa_o = a^{-1}$. We can discretize the bending energy as

$$E_b[\theta] = \frac{B}{2} \sum_0^n l \left( \frac{\theta^{n+1} - \theta^n}{l} - \kappa_o \right)^2 .$$

Assuming