# Hand Worked Bayesian Thermostat via Active Inference

Fraser Paterson

May 2, 2023

Note: I've lifted much of the following explanations from the paper: "Simulating Active Inference Processes by Message Passing", found here: `https://www.frontiersin.org/articles/10.3389/frobt.2019.00020/full`

This is the paper that contains the most relevant example of the "Bayesian Thermostat" problem. The paper was concerned with a demonstration on how to accomplish Active Inference on a Forney factor graph, so some of the following explanation choices might owe to the fact that this is the context in which they were conceived. The factor graph method is entirely incidental to the example Active Inference solution implementation for the "Bayesian Thermostat" problem.

## 1 Introduction

The agent and environment are modeled as dynamical systems with hidden state variables.

The environment executes a process: $(y_t, z_t) = R_t(z_{t-1}, a_t)$ where $a_t$ is an action, $z_t$ is a latent (hidden) state vector and $y_t$ is the output signal. $R_t$ is the transition function.

The agent maintains a *generative model* of the environmental process. This is specified as a probabilistic model:

$$p_t(x, s, u)$$

where $x$, $s$ and $u$ are sequences of observations, internal states and controls (respectively).

the action at time $t$ in the environmental process, is represented in the agent's generative model by a control variable: $u_t$ and an environmental output: $y_t$ by the sensory variable: $x_t$. $x_t$ is also called an "observation".

The agent has a single goal, to minimize the free energy. This roughly corresponds to minimizing the precision-weighted prediction errors for its sensory inputs: $x$.

Minimizing the free energy leads to posterior beliefs over both the states $s$ (which affects predictions and thereby prediction errors) and controls $u$. The controls are observed by the environment as actions that lead to changes in the sensory inputs. these changes in sensory inputs may also affect precision errors. Hence inference over the states and the controls both minimize the free energy.

Since Active Inference reasons from observations toward controls, the inference process requires the definition of an "inverse" probabilistic model, sometimes called the "recognition model" or the "approximate posterior": $q_t$. Full Bayesian inversion of $p_t$ is usually intractable so epistemically bounded agents are modeled as approximating "full" Bayesian inference my means of the minimization of a variational free energy functional defined thus:

$$
\begin{aligned}
F[q_t] &= \int_{q_t} q_t(s,u) \log \frac{q_t(s,u)}{p_t(x,s,u)} && \text{(1a)} \\
&= \underbrace{-\log p_t(x)}_{\text{surprise}} + \underbrace{\int_{q_t} q_t(s,u) \log \frac{q_t(s,u)}{p_t(s,u|x)}}_{\text{posterior KL divergence}} && \text{(1b)}
\end{aligned}
$$

From $p_t(x,s,u) = p_t(s,u|x)p(x)$. Now minimizing 1 renders the free energy an upper bound on the suprisal: $-\log p_t(x)$ while also approximating the (generally unavailable) true posterior: $p_t(s,u|x)$ with the approximate posterior: $q_t$.

We have seen that the agent's internal, generative model $p_t$ expresses the agent's prior beliefs about how the environmental process generates observations from actions. Now I'll explain a simulation protocol for online Active Inference.

Consider an agent with a state-space model factorization for its internal model, given by:

$$
p_t(x,s,u) \propto p(s_{t-1}) \prod_{k=t}^{t+T} \underbrace{p(x_k|s_k)}_{\text{observation}} \underbrace{p(s_k|s_{k-1},u_k)}_{\text{state transition}} \underbrace{p(u_k)}_{\text{control}} \underbrace{p'(x_k)}_{\text{target}} \tag{2}
$$

where $p'(x_k)$ are target priors over desired future outcomes. W call a sequence of future controls $u = (u_t, u_{t+1}, ..., u_{t+T})$ a *policy.* Through inference, the posterior over policies becomes dependent on the hidden state sequence $s$. Prior to inference, the model requires the definition of a prior belief over policies that constrain attainable control states.

We assume the agent has access to a variational distribution, the recognition distribution:

$$
q_t(x,s,u) \tag{3}
$$

which encodes the agent's posterior beliefs about all latent states. Since the future is necessarily unobserved, the recognition distribution includes the future observation variables as well. There is a distinction between the agent's prior "generative" beliefs $p_t(x,s,u)$ and the agent's posterior "recognition" beliefs $q_t(s,s,u)$, which we will finesse below.

At the start of the simulation, say $t = 1$ we will set $q_t$ to be an uninformative distribution. As time progresses, observations become available and subsequent inferences will tune $q_t$ to an ever more informed distribution. Often the recognition distribution is assumed to be fully factorized (mean-field assumption).

In general we may write $q_t(x, s, u) = q(s, s|u)q(u)$. Since actions are real-valued, we generally enforce a deterministic posterior belief over policies:

$$q(u_t, ..., u_{t+T}) = \prod_{k=t}^{t+T} \delta(u_k - \upsilon_k) \tag{4}$$

Where $\upsilon_k$ are parameters to be determined in the free-energy minimization process. In other words, while the prior beliefs over policies $p(u)$ may contain uncertainties, we will fix the posterior belief over policies $q(u)$ on a single concrete sequence.

As time progresses, at each step, the agent interacts with its environment through exchanging actions and observations followed by processing on these observations. Everything the agent does can be described as updates to its internal and recognition models - $p_t$ and $q_t$.

there are three phases per timestep: (1) act-execute-observe, (2) infer (process observations), and (3) "slide", in which we move one time slice ahead. See 1.

**Algorithm 1** Online Active Inference Algorithm

```
Require: $p_1(x, s, u)$ and $q_1(x, s, u)$ // model specification
1: for $t = 1$ to $\infty$ do
2:     Act-Execute-Observe // update to $p_t(x, s, u)$
3:     Infer // update to $q_t(x, s, u)$
4:     Slide // move one time slice ahead
5: end for
```

## 2 Act-Execute-Observe Step

Since we assumed that the posterior beliefs over control states were constrained by $\delta(u_k - \upsilon_k)$, we will select the action at time $t$ as

$$a_t = \upsilon_t \tag{5}$$

The agent's posterior belief $q(u_t) = \delta(u_t - \upsilon_t)$ will be passed over to the control state $u_t$ to the action, leading to $a_t = \upsilon_t$. Next, this action is imposed by the agent, onto the environmental process $R_t$, generating outputs $y_t$ via:

$$(y_t, z_t) = R_t(z_{t-1}, a_t) \tag{6}$$

Where $z_t$ are the dynamic states of the environmental process. We call the environmental processing the *execution* phase.

## 3 Infer Step

The initial model will have changed as a result of acting and observing. In the infer step, the consequences of these changes are processed, for the model's latent variables. We update the approximate posterior from $q_{t-1}$ to $q_t$ by free energy minimization. Generally, the approximate posterior is parameterized by sufficient statistics $\mu$ such that we can write: $q_t(x, s, u) = q(x, s, u|\mu_t)$. Free energy minimization then amounts to finding $\mu_t$ via:

$$\mu_t = \arg\min_{\mu} \int q(x, s, u|\mu) \log \frac{q(x, s, u|\mu)}{p_t(x, s, u)} \, dx \, ds \, du \tag{7}$$

## 4 Slide Step

I'm not ultimately sure how relevant the three "steps" are to the bare bones operation of non-factor graph based Active Inference. See the paper for detail on this step: `https://www.frontiersin.org/articles/10.3389/frobt.2019.00020/full`. These three steps are not essential to the Bayesian Thermostat example per se, so I think I can omit them.

## 5 Bayesian Thermostat

In this example implementation, an agent can move along a line in a temperature gradient field. The agent aims to find the position on the line that yields a desired temperature, relative to a heat source.

The environmental process is modeled as:

$$T(z) = \frac{T_0}{z^2 + 1} \tag{8}$$

Where the temperature $T$ is a function of the agent's position $z$, where this is simply the agent's displacement from the origin. $T_0$ is the temperature at the origin. We'll assume $T_0 = 100.0$.

The environmental process is guided by actions $a_t$ which reflect the velocity of the agent. The output of the environmental process is the temperature observed by the agent. We'll assume the agent observes a noisy temperature, leading to the following equations for the environmental process:

$$z_t = z_{t-1} + a_t \tag{9a}$$

$$y_t \sim \mathcal{N}(T(z_t), \theta) \tag{9b}$$

We'll use an initial position of $z_0 = 2$ and an observation noise variance: $\theta = 10^{-2}$.

## 5.1 Internal Model Specification

The agent's internal model follows the factorization of equation 2. The goal prior encodes a belief about a preferred temperature $x_\phi = 4$, as

$$p'(x_k) = \mathcal{N}(x_k | x_\phi, \theta) \tag{10}$$

We'll endow the agent with an accurate model of system dynamics

$$p(s_k | s_{k-1}, u_k) = \mathcal{N}(s_k | s_{k-1} + u_k, \theta). \tag{11}$$

However, in order to challenge the agent, we hamper the observation model. Instead of the actual temperature profile (Equation 9b), we use

$$p(x_k | s_k) = \mathcal{N}(x_k | -s_k, \theta) \tag{12}$$

which simply specifies that the observed temperature decreases with position. The internal model definition is completed by specifying a prior for controls and a vague prior for the initial state:

$$p(u_k) = \mathcal{N}(u_k \mid 0, \theta) \tag{13a}$$

$$p(s_0) = \mathcal{N}(s_0 \mid 0, 10^{12}) \tag{13b}$$

Substituting Equations (10-13) in Equation (2) then returns the full internal model definition.

## 5.2 Simulation Execution

I think I'll need to now walk through a few iterations of Algorithm (1). I'll use the notebook from: `https://github.com/biaslab/RxAgents/blob/main/Thermostat/Active%20Inference%20Bayesian%20thermostat.ipynb` to "translate" from the RxInfer.jl implementation to an hand-written example with equations.