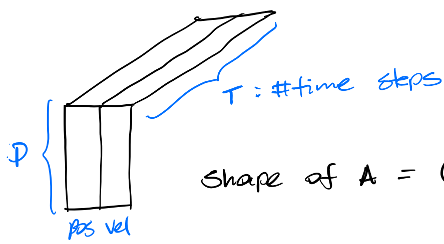
Latent System Components Updates

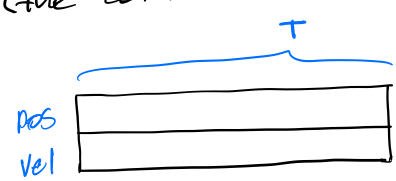
Variables

- A : tensor containing the latent values of person $1, 2, 3, \ldots, P$ over time (e.g. a numerical value or vector summarizing someone's vocal style or cognitive state). In a 2D component, each value/embedding is a 2D vector containing position and velocity.

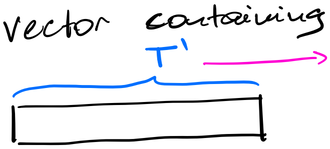


Shape of $A = (P, 2, T)$

- X : matrix similar to A but with a single subject/entity (the common cause).



- c : vector containing values of coordination over time.



Does not have to match T. In the code we index c to retrieve a subvector containing only the c values that match the times in A's time scale. The number of timesteps in this subvector of c is T.

Updates

We update A_t^p using a modified Newtonian dynamics.

subject p

Traditional Newtonian dynamics

$$\begin{cases} Pos_t^p = Pos_{t-1}^p + Vel_{t-1}^p \\ Vel_t^p = Vel_{t-1}^p \end{cases}$$

→ constant coordination

Modified Newtonian dynamics

Eq (I)

$$\begin{cases} Pos_t^p = Pos_{t-1}^p + Vel_{t-1}^p \\ Vel_t^p = (1-C_t)Vel_{t-1}^p + C_t\left(\dfrac{\sum_{p' \in P-\{p\}} Vel_{t-1}^{p'}}{P-1}\right) \\ Vel_t^p = (1-C_t)Vel_{t-1}^p + C_t Vel_t^x \end{cases}$$

(w/o common cause)

→ Average previous velocity of the other subjects.

(w/ common cause)

→ Velocity of the common cause at the current time t not t-1. The common cause affects subjects synchronously.

Example

For sampling, it's fine to use loops. For inference, we want to avoid them for fast computation. So we are going to develop our updates as matrix/tensor operations.

In the log-prob function, we get a sample of the variables in the model and we need to compute log-probabilities according to our update rules. So the values of A, C, and X will be given (a sample)

Let $P=3$ and $T=3$.

$$A = \begin{array}{c} \overbrace{\hphantom{a_0 \quad a_1 \quad a_2}}^{T} \\ \left.\begin{bmatrix} a_0 & a_1 & a_2 \\ b_0 & b_1 & b_2 \\ c_0 & c_1 & c_2 \end{bmatrix}\right\} P \end{array}, \text{ where } a_t = [pos_t^a, vel_t^a]$$

$$X = \begin{bmatrix} pos_0^x & pos_1^x & pos_2^x \\ vel_0^x & vel_1^x & vel_2^x \end{bmatrix}$$

1) Non-common cause case.

With the auxiliary matrix S below, for each subject p, we can create a tensor of entries comprised by the sum of the values of the others. E.g. $A[0] = A[1] + A[2]$.

$$S = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

Multiplying S with A along the second dimension, we get $\Sigma = \frac{SA}{2} = \frac{1}{2}\begin{bmatrix} b_0+c_0 & b_1+c_1 & b_2+c_2 \\ a_0+c_0 & a_1+c_1 & a_2+c_2 \\ a_0+b_0 & a_1+b_1 & a_2+b_2 \end{bmatrix}$ → We will use this for the second part of Eq I.

Let's focus on the matrix $A[0] = \begin{bmatrix} pos_0^a & pos_1^a & pos_2^a \\ vel_0^a & vel_1^a & vel_2^a \end{bmatrix}$ for a moment. Let's define the following auxiliary tensors

$$F = [F_0, F_1, F_2]$$

$$\left[\begin{bmatrix} 1 & 1 \\ 0 & (1-c_1) \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 0 & (1-c_2) \end{bmatrix} \right]$$

$$U = [U_0, U_1, U_2]$$

$$\left[\begin{bmatrix} 0 & 0 \\ 0 & c_1 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & c_2 \end{bmatrix} \right]$$

↳ we start at index 1

Now we can use these matrices to update $A[0]$

$$\hat{A}[0] = \underbrace{F A[0] + U \Sigma[0]}_{\text{over the time axis}} = \begin{bmatrix} \underbrace{\begin{matrix} pos_0^a + vel_0^a \\ (1-c_1) vel_0^a + \frac{c_1}{2}(vel_0^b + vel_0^c) \end{matrix}}_{\hat{A}_1[0]} & \underbrace{\begin{matrix} pos_1^a + vel_1^a \\ (1-c_2) vel_1^a + \frac{c_2}{2}(vel_1^b + vel_1^c) \end{matrix}}_{\hat{A}_2[0]} \end{bmatrix}$$

Notice this is the matrix with the update equations for $A[0]$. We can now compare this (expected) with the original sample. Ultimately, the inference will learn how to generate better samples such that $A_{1:T}[i] \approx \hat{A}[i] \ \forall i \in P$ or $A_{1:T} \approx \hat{A}$.

We want to do the same for the other subjects (A[1] and A[2]). It turns out we can do $FA + U\Sigma$ directly provided that we iterate over the correct axes.

Note: We skipped $t=0$ which we can compute as $A_0 \approx \mu_{a_0}$
Initial mean.

So, $\text{log-prob} = \mathcal{N}(A_0; \mu_{a_0}, \sigma_a^2)$
$\quad\quad += \mathcal{N}(A_{1:T}; \hat{A}, \sigma_a^2)$

2) Common cause case

We can just replace Σ with X but theres a caviat. In the common cause case X affects A at the same time step. So,

$$FA[0] + UX_{1:T} = \begin{bmatrix} Pos_0^a + vel_0^a & Pos_1^a + vel_1^a \\ (1-c_1)vel_0^a + c_1 vel_1^x & (1-c_2)vel_1^a + c_2 vel_2^x \end{bmatrix}$$

We can use X[None, :].repeat(3, axis=0) such that X has the same shape as Σ and the computation works seamlessly.

For $t=0$, we have a blending in this case still but between μ_{a_0} and vel^x. So, an easier way to do that is to concatenate μ_{ao} with A, and do just the blending:

$$FA[0] + UX_{0:T} = \begin{bmatrix} \mu_{ao}[0] + \mu_{ao}[1] \\ (1-c_0)\mu_{ao}[0] + c_0 vel_0^x \end{bmatrix} \quad \boxed{pos_0^a + vel_0^a}$$

now we use 0

Same as the previous matrix

No need to do that