2.3 TRAINING

All model variables except \( S \) are observed during training. Because all other variables are conditionally independent of \( S \) given \( P \) and \( P \) is observed, we can factor the full model into a latent variable model composed of \( S \) and \( P \), and a fully observed model containing the rest of the parameters. We can learn the parameters of \( S \) and \( P \) separately from the rest, simplifying training.

The production process is a hidden Markov model (HMM) (Rabiner, 1990), where \( S_t \) is the latent state and \( P_{t}^{1}, \ldots, P_{t}^{N} \) are the emissions. The parameters of the HMM are the initial state probabilities \( P(S_0) = \text{Multinomial}(\eta_1, \ldots, \eta_M) \), the state transition probabilities \( P(S_t | S_{t-1} = s) = \text{Multinomial}(\pi_1^s, \ldots, \pi_M^s) \), and the inflated Poisson parameters \( P(P_t^i = k > 0 | S_t = s) = \nu_s^i \cdot \text{Pois}(k - 1; \lambda_i^s) \).

At training time, we observe \( P_t \). We can then estimate \( \Phi \) in the usual way using the Expectation Maximization (EM) algorithm. We initialized the EM algorithm as follows: the \( \eta \) and \( \pi \) parameters are set to \( 1/M \), values of \( \nu \) are drawn from a Uniform(0, 1), and \( \lambda \) parameters are drawn from a Uniform(0, 10).

The “unobserved loss” probabilities \( \ell^i \) are estimated as the number of unobserved losses of units of type \( i \) divided by the number of unit-epochs (analogous to human-years) during which a unit of that type existed.

For unit types that were present in at least 100 unit-epochs, \( \ell^i \) was estimated using additive smoothing as \( \hat{\ell}^i = \frac{d^i + 1}{\hat{N} + N} \) where \( d^i \) is the number of unobserved losses of unit type \( i \) and \( \hat{N} \) is the number of unit-epochs for type \( i \). The smoothing ensures that all unit types have non-zero \( \ell^i \) even if there were no unobserved losses in the training data. For types that were not present in at least 100 unit-epochs, the median estimate was used.

The functions \( \hat{\ell}^i \) giving the parameters of the distributions of \( O_{t}^{i} \) are learned via logistic regression with a maximum likelihood objective using the R package aod (Lesnoff et al., 2010). We fit a \( \hat{\mu}^i \) parameter for unit type \( i \) only if a unit of that type was observed on at least 100 occasions in the dataset. We fit a \( \hat{\rho}^i \) parameter only if the unit type met the condition for \( \hat{\mu}^i \) and there were at least two of the unit type present (but not necessarily observed) on at least 100 occasions. The reason for the condition on \( \hat{\rho} \) is that if it is rare for more than one instance of the unit to exist, then there is little dispersion in the data, and the estimate of \( \hat{\rho}^i \) will be near 0. In this case, \( \hat{\rho}^i \) would merely be modeling the tendency not to build more than one unit, which is properly the job of \( P_t \). For types that did not have enough data for \( \hat{\mu} \) or \( \hat{\rho}, \) the median of the estimated regression coefficients are used.

2.4 INFERENCE

We denote the subset of latent variables for a slice \( t \) as \( X_t = \{ S_t, P_t^1, \ldots, P_t^N, U_t^1, \ldots, U_t^N \} \), and the observed variables as \( Y_t = \{ E_t, K_t^1, \ldots, K_t^N, O_t^1, \ldots, O_t^N \} \). We use the lowercase \( y_t \) and \( x_t \) to denote instantiations of these variables (i.e., \( y_t \) refers to the evidence at time \( t \)). Because each \( U_t^i \) is conditioned on \( S_t \), and \( S_t \) and \( U_t^i \) are Markovian, an exact filtering pass would require representing the forward message \( \alpha_t = P(S_t, U_t^1, \ldots, U_t^N) \). This is intractable for even a modest number of types, since the size of the joint distribution is \( M^{N_{\text{max}}} \). However, a key observation is that, given the history of the strategy state, \( S_{0:t} = (S_0, \ldots, S_t) \), the model up to time \( t \) decomposes into \( N \) independent HMMs, each tracking the count of a single type. We leverage this structure by employing a Rao-Blackwellized particle filter (RBPF) for approximate inference (Doucet et al., 2000; Murphy, 2000).

In our application of RBPF, we draw particles of \( S_{0:t} \), and compute \( P(U_t^i | S_{0:t}) \) analytically via standard HMM filtering. Following an importance sampling framework, particles are generated at each time step from a proposal distribution \( Q(S_t) \). We use the state transition model for our proposal, \( Q(S_t) = P(S_t | s_{t-1}) \). While this choice ignores recent evidence at time \( t \), it is computationally efficient to sample, and there are often periods of no evidence, anyway.

At \( t = 0 \) we draw \( R \) particles from the initial state prior \( s_{0}^{1}, \ldots, s_{0}^{R} \sim P(S_0) \). Each particle has an importance weight \( w_t^r = \phi(s_t^r) / Q(s_t^r) \), where \( \phi(s_t^r) \) is the probability of the particle’s value \( s_t^r \) given by the full model (up to normalization). At \( t = 0 \), \( w_0^r = (P(y_0 | s_t^r) P(s_t^r) / Q(s_t^r) \). For \( t > 0 \), each particle generates its next value of the state \( s_{t}^r \sim Q(S_t) = P(S_t | s_{t-1}^r) \). We then update its weight using the ratio:

\[
\hat{w}_t^r = \frac{P(Y_t = y_t | s_t^r, s_{0:t}, s_{0:t-1}^r) P(S_t = s_t^r | s_{t-1} = s_{t-1}^r)}{P(S_t = s_t^r | s_{t-1} = s_{t-1}^r)}
\]

The new weight for particle \( r \) is then \( \hat{w}_t^r = w_t^{r-1} \hat{w}_t^r \).

Because our proposal distribution is identical to \( P(S_t | s_{t-1}) \) (canceling out the denominator), we are only interested in the likelihood term of the numerator, \( P(Y_t = y_t | s_{t-1} = s_t^r, s_{0:t-1}^r) \), which factors as

\[
\prod_{i=1}^{N} \left[ P(U_t^i | U_{t-1}^i, P_t^i, K_{t-1}^i) P(U_{t-1}^i) \right]
\]

\[
\cdot P(P_t^i | S_t = s_t^r) P(O_t^i | U_t^i, E_t^i)
\]

\( P(U_{t-1}^i) \) is a forward-pass message that captures the posterior marginal distribution over the counts of unit type \( i \) at time \( t - 1 \). After we weight a sample, we