

# Experience-Weighted Attraction

## General Principles

Social learning models often aim to disentangle asocial learning (learning from personal experience) from social learning (learning from others). The **Experience-Weighted Attraction (EWA)** model Barrett, McElreath, and Perry (2017) provides a robust framework for this. In the context of the Panama capuchin monkey study Barrett, McElreath, and Perry (2017), the model tracks how individuals update their “attractions” to different foraging techniques based on their own success and the social cues they observe from others.

The core idea is that individuals have a set of attractions  $A_{i,j,t}$  for each technique  $j$ . These attractions are updated over time, and the probability of choosing a technique is a mixture of asocial preferences (derived from attractions) and social influence.

In this documentation, we present two primary variations of the EWA model adapted from Barrett, McElreath, and Perry (2017):

1. **Social Learning (EWA)**: A streamlined version (Time Frequency Pay-off model) focusing on the interplay between individual experience (asocial learning) and social frequency biased by payoffs. It uses a reduced parameter set (4 varying effects) to identify the core drivers of behavioral diffusion.
2. **Social Learning (EWA-ILV)**: A comprehensive hierarchical model (Experience-Weighted Attraction with Individual Level Covariates) that evaluates multiple social biases (e.g., kinship, prestige, age similarity) and explicitly models how an individual’s age influences their learning rate ( $\phi$ ) and reliance on social cues ( $\gamma$ ).

---

## 1. Social Learning (EWA)

The “EWA” model is a basic version that includes asocial learning and pay-off biased social learning. It evaluates four varying effects: learning rate, social weight, conformity, and pay-off bias.

## Example

### **i** Note

- **Hierarchical Structure:** Intercepts for learning rates and social influence vary by individual (`id`). Barrett, McElreath, and Perry (2017) use a Cholesky-factored multivariate normal prior to account for correlations between these effects.
- **Dtype Sensitivity:** Ensure all data arrays are `float64` to match JAX's 64-bit backend requirements for numerical stability.

## Python

```
from BI import bi
import jax.numpy as jnp
import jax

m = bi(platform='cpu')

def model(K, J, tech, id, bout, y_prev, s, ps):
    # Priors
    lam = m.dist.exponential(name="lambda", rate=1.0)
    mu = m.dist.normal(name="mu", loc=jnp.zeros(4), scale=1.0)
    sigma = m.dist.exponential(name="sigma", rate=jnp.full(4, 3.0))
    L_Rho = m.dist.lkj_cholesky(name="L_Rho", dimension=4, concentration=3.0)
    z = m.dist.normal(name="z", loc=jnp.zeros((4, J)), scale=1.0)

    # Varying effects
    L_scaled = L_Rho * sigma[:, None]
    a_id = (L_scaled @ z).T

    # Parameters per individual
    phi_arr = jax.nn.sigmoid(mu[0] + a_id[id, 0]) # Learning rate
    gamma_arr = jax.nn.sigmoid(mu[1] + a_id[id, 1]) # Social weight
    fconf_arr = jnp.exp(mu[2] + a_id[id, 2]) # Conformity
    Bpay_arr = mu[3] + a_id[id, 3] # Pay-off bias

    # Scan carry: Attractions
    def step_fn(carry, x):
        AC = carry
        (phi_i, gamma_i, fconf_i, Bpay_i, tech_i, id_i, bout_i, prev_y_i, s_i, ps_i) = x
```

```

# Update attractions (EWA)
ac_new = jnp.where(bout_i == 1, jnp.zeros(K), (1.0 - phi_i) * AC[id_i] + phi_i * prev

# Asocial choice probability
prob_individual = jax.nn.softmax(lam * ac_new)

# Social choice probability (Pay-off bias)
lin_mod = jnp.concatenate([jnp.ones(1), jnp.exp(Bpay_i * ps_i[1:])]) * jnp.power(s_i
safe_denom = jnp.where(jnp.sum(lin_mod) > 0, jnp.sum(lin_mod), 1.0)
prob_social = lin_mod / safe_denom

# Mixture probability
prob_mix = (1.0 - gamma_i) * prob_individual + gamma_i * prob_social
prob_final = jnp.where(bout_i > 1, prob_mix, prob_individual)

return AC.at[id_i].set(ac_new), prob_final

AC_init = jnp.zeros((J, K))
xs = (phi_arr, gamma_arr, fconf_arr, Bpay_arr, tech, id, bout, y_prev, s, ps)
_, probs_matrix = jax.lax.scan(step_fn, AC_init, xs)

m.dist.categorical("obs", probs=probs_matrix, obs=tech)

m.fit(model)
m.summary()

```

## R

```

library(BayesianInference)
m <- importBI(platform='cpu')

model <- function(K, J, tech, id, bout, y_prev, s, ps) {
  # Priors
  lam <- m$dist$exponential(name="lambda", rate=1.0)
  mu <- m$dist$normal(name="mu", loc=jnp$zeros(4L), scale=1.0)
  sigma <- m$dist$exponential(name="sigma", rate=jnp$full(4L, 3.0))
  L_Rho <- m$dist$lkj_cholesky(name="L_Rho", dimension=4L, concentration=3.0)
  z <- m$dist$normal(name="z", loc=jnp$zeros(tuple(4L, J)), scale=1.0)

  # ... Equivalent logic to Python implementation ...
}

```

```
m$fit(model)
m$summary()
```

## Julia

```
using BayesianInference
m = importBI(platform="cpu")

@BI function model(K, J, tech, id, bout, y_prev, s, ps)
    # Priors
    lam = m.dist.exponential(name="lambda", rate=1.0)
    mu = m.dist.normal(name="mu", loc=jnp.zeros(4), scale=1.0)
    sigma = m.dist.exponential(name="sigma", rate=jnp.full(4, 3.0))
    L_Rho = m.dist.lkj_cholesky(name="L_Rho", dimension=4, concentration=3.0)
    z = m.dist.normal(name="z", loc=jnp.zeros(4, J), scale=1.0)

    # ... Equivalent logic to Python implementation ...
end
m.fit(model)
m.summary()
```

## Mathematical Details

The “Experience-Weighted Attraction” (EWA) model links individual learning rates and social frequency to behavioral choice. Throughout this section, we use a consistent three-part index:  $i$  denotes the individual,  $k \in \{1, \dots, K\}$  denotes the technique, and  $t$  denotes the observation (time step).

### 1. Mixture Likelihood

At each observation, the probability of choosing a single technique from  $K$  options is expressed through a *Categorical* likelihood:

$$Y_{[i,t]} \sim \text{Categorical}(\theta_{[i,t]})$$

where:

- $Y_{[i,t]}$  is the technique chosen by individual  $i$  at time step  $t$ .
- $\theta_{[i,t]}$  is a probability vector of length  $K$  giving the probability of each technique for individual  $i$  at time step  $t$ .

Unlike a standard Categorical model where  $\theta$  is fixed for a group of observations, here  $\theta_{[i,t]}$  is a **different vector at every time step** for every individual, because the attraction scores and social cues update sequentially.

The probability that individual  $i$  chooses technique  $k$  at time  $t$  decomposes as:

$$\begin{aligned}\theta_{[i,k,t]} &= (1 - \gamma_i) I_{[i,k,t]} + \gamma_i S_{[i,k,t]} \\ \gamma_i &= \text{sigmoid}(\mu_2 + a_{id}[i, 2])\end{aligned}$$

Where:

- $I_{[i,k,t]}$  is the asocial choice probability: derived from individual  $i$ 's personal experience with technique  $k$ .
- $S_{[i,k,t]}$  is the social choice probability: derived from observing conspecifics use technique  $k$ .
- $\gamma_i \in [0, 1]$  is individual  $i$ 's social learning weight: at 0 the individual relies entirely on personal experience; at 1 entirely on social information. It is derived from the hierarchical prior (see §2):

## 2. Asocial Component — $I_{[i,k,t]}$

The asocial probability is a softmax over accumulated attraction scores — it converts raw attraction values into a proper probability distribution over techniques:

$$\begin{aligned}I_{[i,k,t]} &= \frac{\exp(\lambda \cdot A_{[i,k,t]})}{\sum_{k'=1}^K \exp(\lambda \cdot A_{[i,k',t]})} \\ \lambda &\sim \text{Exponential}(1)\end{aligned}$$

The numerator scores technique  $k$  by the individual's scaled attraction to it. The denominator sums this over all  $K$  techniques ( $k' = 1, \dots, K$ ), ensuring the probabilities sum to 1 and that each technique is evaluated relative to all alternatives.

Where:

- $A_{[i,k,t]}$  is the attraction score of individual  $i$  for technique  $k$  at time  $t$ . It is a running exponentially weighted average of the personal payoffs individual  $i$  has obtained from technique  $k$  across past bouts. Larger values indicate a stronger learned payoff expectation.
- $A_{[i,k',t]}$  in the denominator ranges over the attraction scores for all  $K$  techniques, making choice proportional to each option's scaled attraction relative to the full set.
- $\lambda > 0$  is the multinomial sensitivity (inverse temperature): higher values make choice more deterministic toward the highest-attraction option;  $\lambda \rightarrow 0$  yields random choice. The Exponential(1) prior keeps  $\lambda$  positive and places most mass near zero, expressing a prior expectation that choice is moderately stochastic. Larger values are possible but require evidence from the data.

Attraction scores update between foraging bouts via the EWA rule:

$$A_{[i,k,t+1]} = (1 - \phi_i) A_{[i,k,t]} + \phi_i p_{[i,k,t]}$$

$$\phi_i = \text{sigmoid}(\mu_1 + a_{id}[i, 1])$$

$$\mu \sim \mathcal{N}(\mathbf{0}_4, 1)$$

$$\sigma \sim \text{Exponential}(3)$$

$$\mathbf{L}_\rho \sim \text{LKJCholesky}(4, \eta = 3)$$

$$\mathbf{Z} \sim \mathcal{N}(\mathbf{0}_{4 \times J}, 1)$$

$$\mathbf{a}_{id} = (\mathbf{L}_\rho \cdot \text{diag}(\sigma) \cdot \mathbf{Z})^\top$$

Where:

- $A_{[i,k,t]}$  is the attraction score of individual  $i$  for technique  $k$  at time  $t$ .
- $p_{[i,k,t]}$  is the personal yield (payoff) individual  $i$  obtained from technique  $k$  at time  $t$  — concretely, a measure of food-extraction efficiency (e.g. rate of fruit opening).
- $\phi_i \in [0, 1]$  is individual  $i$ 's attraction updating weight. Near 1: attractions closely track the most recent payoff. Near 0: strong memory of cumulative past experience. It is constructed from a multivariate hierarchical prior shared by all four individual-level learning parameters  $(\phi_i, \gamma_i, f_i, \beta_i)$
- $\mu \in \mathbb{R}^4$  are population-level means
- $a_{id}[i, \cdot]$  are individual-level correlated deviations reconstructed via non-centered parameterization
- $\sigma \in \mathbb{R}_+^4$  are the per-parameter standard deviations (shrinkage parameters)
- $\mathbf{L}_\rho$  is the Cholesky factor of the  $4 \times 4$  correlation matrix: the LKJ prior ( $\eta = 3$ ) moderately shrinks correlations toward zero.
- $\mathbf{Z} \in \mathbb{R}^{J \times 4}$  is a matrix of standard normal deviates (one row per individual).

### 3. Social Component — $S_{[i,k,t]}$

The social probability combines payoff bias with frequency-dependent conformity:

$$S_{[i,k,t]} = \frac{N_{[i,k,t]}^{f_i} \exp(\beta_i \pi_{[i,k,t]})}{\sum_{k'=1}^K N_{[i,k',t]}^{f_i} \exp(\beta_i \pi_{[i,k',t]})}$$

$$f_i = \exp(\mu_3 + a_{id}[i, 3])$$

$$\beta_i = \mu_4 + a_{id}[i, 4]$$

Where:

- $N_{[i,k,t]}$  is the *frequency-bias cue*: the number of times individual  $i$  observed conspecifics performing technique  $k$  at time  $t$ .
- $f_i > 0$  is individual  $i$ 's conformity exponent.  $f_i > 1$ : positive frequency-dependent bias (disproportionate copying of the most common technique);  $f_i = 1$ : purely linear frequency weighting;  $f_i < 1$ : tendency to avoid the majority. The exponential link ensures  $f_i > 0$ ; the prior  $\mu_3 \sim \mathcal{N}(0, 1)$  corresponds to  $f \approx 1$  a priori.
- $N_{[i,k',t]}^{f_i}$  in the denominator ranges over all  $K$  techniques with the same conformity exponent  $f_i$ , normalising the social probabilities.
- $\pi_{[i,k,t]}$  is the payoff observed from social demonstrators using technique  $k$  at time  $t$ .
- $\beta_i$  is individual  $i$ 's payoff-bias coefficient: positive values mean the individual preferentially copies techniques demonstrated with higher payoff. It is on the unconstrained real scale; the prior  $\mu_4 \sim \mathcal{N}(0, 1)$  expresses prior uncertainty about the direction of payoff bias.

The numerator scores technique  $k$  by two multiplicative components: how often it was observed being used by conspecifics ( $N_{[i,k,t]}^{f_i}$ , frequency bias) and how profitable it appeared in their hands ( $\exp(\beta_i \pi_{[i,k,t]})$ , payoff bias). The denominator ( $k' = 1, \dots, K$ ) sums these scores over all techniques to normalise to a probability. Together they implement the idea that individuals copy techniques that are simultaneously common *and* profitable.

---

## 2. Social Learning (EWA-ILV)

The “EWA-ILV” model expands the EWA version by incorporating a wider array of social biases (kinship, prestige, age similarity, etc.) and modeling the linear influence of individual age on the learning parameters  $\phi$  and  $\gamma$ .

### Example

### Python

```
from BI import bi
import jax.numpy as jnp
import jax

m = bi(platform='cpu')

def model(K, J, tech, id, bout, y_prev, s, ps, ks, pr, co, yo, age):
    # Priors (8 varying effects)
```

```

lam = m.dist.exponential(name="lambda", rate=1.0)
mu = m.dist.normal(name="mu", loc=jnp.zeros(8), scale=1.0)
sigma = m.dist.exponential(name="sigma", rate=jnp.full(8, 3.0))
L_Rho = m.dist.lkj_cholesky(name="L_Rho", dimension=8, concentration=3.0)
z = m.dist.normal(name="z", loc=jnp.zeros((8, J)), scale=1.0)
b_age = m.dist.normal(name="b_age", loc=jnp.zeros(2), scale=1.0)

# Varying effects
L_scaled = L_Rho * sigma[:, None]
a_id = (L_scaled @ z).T

# Parameters per individual with age effects
phi_arr = jax.nn.sigmoid(mu[0] + a_id[id, 0] + b_age[0] * age)
gamma_arr = jax.nn.sigmoid(mu[1] + a_id[id, 1] + b_age[1] * age)
fconf_arr = jnp.exp(mu[2] + a_id[id, 2])
B_pay = mu[3] + a_id[id, 3]
B_kin = mu[4] + a_id[id, 4]
B_pres = mu[5] + a_id[id, 5]
B_coho = mu[6] + a_id[id, 6]
B_yob = mu[7] + a_id[id, 7]

# Scan carry: Attractions
def step_fn(carry, x):
    AC = carry
    (phi_i, gamma_i, fconf_i, Bp, Bk, Bpr, Bc, By, t_i, id_i, b_i, py_i, s_i, ps_i, ks_i)

    ac_new = jnp.where(b_i == 1, jnp.zeros(K), (1.0 - phi_i) * AC[id_i] + phi_i * py_i)

    # Asocial choice probability
    prob_individual = jax.nn.softmax(lam * ac_new)

    # Multi-cue social influence
    lin_rest = jnp.exp(Bp * ps_i[1:] + Bk * ks_i[1:] + Bpr * pr_i[1:] + Bc * co_i[1:] + B)
    lin_mod = jnp.concatenate([jnp.ones(1), lin_rest]) * jnp.power(s_i, fconf_i)
    safe_denom = jnp.where(jnp.sum(lin_mod) > 0, jnp.sum(lin_mod), 1.0)
    prob_social = lin_mod / safe_denom

    # Mixture probability
    prob_mix = (1.0 - gamma_i) * prob_individual + gamma_i * prob_social
    has_social_info = jnp.logical_and(b_i > 1, jnp.sum(s_i) > 0)
    prob_final = jnp.where(has_social_info, prob_mix, prob_individual)

```

```

    return AC.at[id_i].set(ac_new), probb_final

AC_init = jnp.zeros((J, K))
xs = (phi_arr, gamma_arr, fconf_arr, B_pay, B_kin, B_pres, B_coho, B_yob,
      tech, id, bout, y_prev, s, ps, ks, pr, co, yo)
_, probs_matrix = jax.lax.scan(step_fn, AC_init, xs)

m.dist.categorical("obs", probs=probs_matrix, obs=tech)

m.fit(model)
m.summary()

```

## R

```
# Equivalent R implementation with 8 effects and age slopes...
```

## Julia

```
# Equivalent Julia implementation with 8 effects and age slopes...
```

## Mathematical Details

The “Experience-Weighted Attraction with Individual Level Covariates” (EWA-ILV) model extends the EWA model in two ways: it incorporates a wider array of social cues in the social learning component, and it allows an individual’s age to shift key learning parameters.

### 1. Mixture Likelihood

The observation likelihood has the same mixture form as the EWA model, using a three-part index: individual  $i$ , technique  $k$ , and time step  $t$ :

$$\theta_{[i,k,t]} = (1 - \gamma_i) I_{[i,k,t]} + \gamma_i S_{[i,k,t]}$$

### 2. Social Component — $S_{[i,k,t]}$

The social probability combines frequency-dependent conformity with a multi-cue log-linear model:

$$S_{[i,k,t]} = \frac{N_{[i,k,t]}^{f_i} \exp(B_{[i,k,t]})}{\sum_{k'=1}^K N_{[i,k',t]}^{f_i} \exp(B_{[i,k',t]})}$$

Where  $B_{[i,1,t]} = 0$  by convention (technique 1 is the aliased baseline), and for  $k > 1$ :

$$B_{[i,k,t]} = \beta_{\text{pay},i} \pi_{[i,k,t]} + \beta_{\text{kin},i} \kappa_{[i,k,t]} + \beta_{\text{rank},i} \zeta_{[i,k,t]} + \beta_{\text{cohort},i} \eta_{[i,k,t]} + \beta_{\text{age},i} \alpha_{[i,k,t]}$$

### 3. Individual-Level Parameters, Age Effects, and Priors

All eight learning parameters are drawn from a correlated multivariate hierarchy. For each individual  $i$ , the varying parameters and deviations are constructed as:

$$\phi_i = \text{sigmoid}(\mu_0 + a_{id}[i, 0] + \beta_\phi \cdot \text{age}_i)$$

$$\gamma_i = \text{sigmoid}(\mu_1 + a_{id}[i, 1] + \beta_\gamma \cdot \text{age}_i)$$

$$f_i = \exp(\mu_2 + a_{id}[i, 2])$$

$$\beta_{\text{pay},i} = \mu_3 + a_{id}[i, 3]$$

$$\beta_{\text{kin},i} = \mu_4 + a_{id}[i, 4]$$

$$\beta_{\text{rank},i} = \mu_5 + a_{id}[i, 5]$$

$$\beta_{\text{cohort},i} = \mu_6 + a_{id}[i, 6]$$

$$\beta_{\text{age},i} = \mu_7 + a_{id}[i, 7]$$

$$\mathbf{a}_{id} = (\mathbf{L}_\rho \text{diag}(\sigma) \mathbf{Z})^\top$$

$$\mu \sim \mathcal{N}(\mathbf{0}_8, 1)$$

$$\sigma \sim \text{Exponential}(\text{rate} = 3)$$

$$\mathbf{L}_\rho \sim \text{LKJCholesky}(8, \eta = 3)$$

$$\mathbf{Z} \sim \mathcal{N}(\mathbf{0}_{8 \times J}, 1)$$

$$\beta_\phi \sim \mathcal{N}(0, 1), \quad \beta_\gamma \sim \mathcal{N}(0, 1)$$

$$\lambda \sim \text{Exponential}(1)$$

Where:

- $N_{[i,k,t]}$  is the frequency cue: the number of times individual  $i$  observed conspecifics performing technique  $k$  at time  $t$ .
- $f_i > 0$  is individual  $i$ 's conformity exponent, constructed with the exponential link to guarantee positive values. The population prior mean  $\mu_2 \sim \mathcal{N}(0, 1)$  corresponds to  $f \approx 1$  (linear frequency copying) a priori.
- $\pi_{[i,k,t]}$  is the payoff observed from social demonstrators using technique  $k$  at time  $t$ . The prior  $\mu_3 \sim \mathcal{N}(0, 1)$  expresses uncertainty on payoff bias direction and magnitude.
- $\kappa_{[i,k,t]}$  is the kin bias cue: whether the demonstrators using technique  $k$  at time  $t$  are matrilineal kin of individual  $i$ . The coefficient  $\beta_{\text{kin},i}$  models the kinship bias, with population prior mean  $\mu_4 \sim \mathcal{N}(0, 1)$ .

- $\zeta_{[i,k,t]}$  is the rank/prestige cue: whether the demonstrators using technique  $k$  at time  $t$  are high-ranking (e.g., alpha) individuals. The coefficient  $\beta_{\text{rank},i}$  models rank bias, with population prior mean  $\mu_5 \sim \mathcal{N}(0, 1)$ .
- $\eta_{[i,k,t]}$  is the cohort cue: the age-similarity between individual  $i$  and demonstrators using technique  $k$  at time  $t$ . The coefficient  $\beta_{\text{cohort},i}$  models cohort bias, with population prior mean  $\mu_6 \sim \mathcal{N}(0, 1)$ .
- $\alpha_{[i,k,t]}$  is the age-bias cue: a year-of-birth prestige bias towards older/more experienced demonstrators using technique  $k$  at time  $t$ . The coefficient  $\beta_{\text{age},i}$  models age-based prestige bias, with population prior mean  $\mu_7 \sim \mathcal{N}(0, 1)$ .
- $\phi_i \in [0, 1]$  and  $\gamma_i \in [0, 1]$  are individual-level attraction-updating and social-learning weights, constructed with the logistic/sigmoid link.
- $\text{age}_i$  is the centered age of individual  $i$ .
- $\beta_\phi$  and  $\beta_\gamma$  are the population-level age slopes on the log-odds scale, determining how an individual's age alters their attraction updating and social learning weight.
- $\mathbf{a}_{id} \in \mathbb{R}^{J \times 8}$  is the matrix of individual-level deviations across the 8 varying parameters, reconstructed using the non-centered parameterization to optimize Hamiltonian Monte Carlo performance.
- $\mathbf{L}_\rho \in \mathbb{R}^{8 \times 8}$  is the Cholesky factor of the correlation matrix, mapping correlations among all 8 parameters.
- $\sigma \in \mathbb{R}_+^8$  represents the standard deviations (shrinkage parameters) for each of the 8 parameters.
- $\mathbf{Z} \in \mathbb{R}^{8 \times J}$  is a matrix of standard normal deviates.
- $\lambda > 0$  is the multinomial sensitivity parameter, governing choice deterministic behavior.

---

## Notes

### **i** Note

- The EWA-ILV model is significantly more complex and requires larger sample sizes to resolve the correlations between varying effects.
- Cholesky decomposition of the correlation matrix is used for both models to ensure positive-definiteness.

## Reference(s)

Barrett, Brendan J., Richard L. McElreath, and Susan E. Perry. 2017. "Pay-Off-Biased Social Learning Underlies the Diffusion of Novel Extractive Foraging Traditions in a Wild Primate." *Proceedings of the Royal Society B: Biological Sciences* 284 (1856): 20170358. <https://doi.org/10.1098/rspb.2017.0358>.