Supporting information for “The evolution of social learning mechanisms and cultural phenomena in group foragers”
Daniel J van der Post¹*, Mathias Franz², Kevin Laland¹
1 Center for Social Learning and Cognitive Evolution, St Andrews University, St Andrews, UK
2 Leibniz Institute for Zoo and Wildlife Research, Berlin, Germany
* E-mail: d.j.vanderpost@gmail.com

Contents

1 Detailed model description ........................................ 2
  1.1 Purpose ......................................................... 2
  1.2 State variables and scales ...................................... 2
  1.3 Process overview and scheduling .............................. 3
  1.4 Design concepts .............................................. 3
  1.5 Specification choices of scaling context ....................... 5
  1.6 Initialization .................................................. 6
  1.7 Input .......................................................... 6
  1.8 Submodels ..................................................... 6
    1.8.1 Forager continuous updates and decision making algorithm .. 6
    1.8.2 Behavior actions of foragers ............................... 7
    1.8.3 Forager discrete updates ................................... 9
    1.8.4 Resources .................................................. 9
  1.9 Comparison to diet model by van der Post & Hogeweg .......... 9

2 Reduction of traditional differences with increasing task difficulty 11
1 Detailed model description

(i) there is a diverse environment with different potential food or resource types of varying quality (Smuts et al., 1987; Chapman and Fedigan, 1990; Altmann, 1998); (ii) resources are limited in their abundance and are depleted when consumed, or contain aversive secondary compounds, such that foragers need to consume a variety of resources (Provenza, 1996); (iii) foragers live in groups with overlapping generations (Smuts et al., 1987); (iv) foragers need to learn about the rewards that are associated with particular resources (Matsuzawa and Hasegawa, 1982, 1983; Birch, 1999; Visalberghi et al., 2003; Hardus et al., 2014); (v) foragers are selective (i.e. do not eat everything they encounter) (Chapman and Fedigan, 1990; Altmann, 1998); (vi) foragers can develop skills associated with processing particular resources and so obtain greater rewards from those resources (Byrne, 1999; Jaeggi et al., 2010); (vii) foragers can observe the foraging choices of their neighbors and this can affect the foragers’ subsequent choices (Visalberghi and Addessi, 2003; Fragaszy and Visalberghi, 2004; Agostini and Visalberghi, 2005; Jaeggi et al., 2010; van de Waal and Whiten, 2012; van de Waal et al., 2014). These empirical observations directly determine the main features of our model, and we study the dynamic processes that they generate. Here we document a full model description based on the ODD protocol for describing simulation models (Grimm et al., 2006).

1.1 Purpose

We develop an event-based, individual-based, spatially extended model of group foragers learning what and how to eat, in order to enable a multi-scale approach to the evolution of social learning mechanisms. Specifically, we study the cultural phenomena generated by evolved social learning mechanisms.

1.2 State variables and scales

Our model consists of a 2-dimensional environment with resource items and foragers. Space and time are continuous (via event-based scheduling), but some variables are updated via discrete time steps where each time point is a minute, a day is 720 minutes (only daylight), and a year is 360 days.

Resource types are described by: a spatial position, a maximum energy value $Q_r$, a task difficulty $H_r$ and $S_r$ which determines how rapidly foragers can initially learn to process a resource type. Specific resource items are also defined by a spatial position and whether they have been consumed this year or not. To generate resource patches, resource items are arranged randomly in a circle around a randomly selected patch center. Resource patches are characterized by: a location, a radius, resource density. The “patch-level” is emergent in the sense that patches can only be observed by foragers by sensing resource items.

Foragers are characterized by the following state variables: age, energy, position, heading, behavior state, behavioral action, time to complete an action (completion time), memory (see below), having a particular feeding target (i.e. a food item), a group identity, and parameters that determine behavior (see Table S1). Forager memory includes: (i) having a particular feeding target (i.e. a food item), (ii) associations $a_{ir}$ between a resource and a reward, (iii) experience $t_{ir}$, the total time spend processing a particular resource type, (iv) a particular resource type for
which the forager has a stimulus for via stimulus enhancement, (v) an estimate of the quality of resources that can be found in the environment $\sigma_{ae}$, and (vi) estimates of how certain foragers are about resources $c_{ir}$. The forager population is fixed, but the number of groups changes when groups die and are 'born'.

1.3 Process overview and scheduling

We distinguish two levels of scheduling: (i) continuous time and (ii) discrete time steps (every minute).

**Discrete updating:** every minute we update individual age, reduce energy due to energy metabolism, determine if an individual dies due to a fixed death rate, starvation or old age, and if so select a parent according to energy levels:

$$\left( \frac{h_i^W}{\sum_{j=1}^{N} h_j} \right)^W$$

where $h_i$ is an individual's energy level, $N$ is the population size and $W(= 3)$ is the selection coefficient. Groups die when the last group member dies, and are born when a group reaches the maximum group size (20 foragers) and is randomly split into two equally sized daughter groups. Mutations in behavior parameters during reproduction generates genetic variation. If variation affects reproductive success, natural selection arises. Each resource type is checked if it is replaced by a new resource type every minute according to $EC$ the rate of environmental change. Every DIG_TIME (100) minutes all consumed resource items are ‘digested’. If any resource type exceeds SATIATION_FRAC*$MAX_STOMACH$ number of items, then the forager attains a temporary satiation aversion for that resource type for SATIATION_TIME minutes.

**Continuous updating:** every behavioral action has a continuous time point at which it is completed (completion time). Actions can therefore end anywhere within a minute (with the resolution of “float” types in C++ programming language), and an action started in a previous minute is completed in the next minute. All foragers are put into a queue according to their completion time. The individual with the lowest completion time is next to complete its action and choose a new action. The completion time represents the duration of an action. Foragers therefore start their actions asynchronously, but action durations can overlap. This is an event-based scheduling. The sequence of actions of a forager is determined by the decision making algorithm.

When foragers consume resource items those items are tagged as ‘consumed’ and become available again the following year. When assessing resources items locally, any consumed resources items are checked whether they have become available again. If so they are tagged as available. This set-up was chosen for computational reasons, such that updating resource items occurs in a ‘need to know’ manner in relation to what foragers are observing locally. As a result we do not need to update all resource items all the time.

1.4 Design concepts

**Emergence:** our model is all about detecting and understanding emerging processes and the interactions between them. The following is a list of emerging phenomena:
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>YEAR</td>
<td>number of days</td>
<td>360</td>
</tr>
<tr>
<td>PATCH_RADIUS</td>
<td>Radius of the patch</td>
<td>10 m</td>
</tr>
<tr>
<td>PATCH_RES</td>
<td>Number of resource types in a given patch</td>
<td>5</td>
</tr>
<tr>
<td>PATCH_RES_PER</td>
<td>Subset of possible resource types in particular patch</td>
<td>1</td>
</tr>
<tr>
<td>N</td>
<td>Number of resources species</td>
<td>250</td>
</tr>
<tr>
<td>M</td>
<td>Population size</td>
<td>100</td>
</tr>
<tr>
<td>W</td>
<td>Selection coefficient which determines the strength of selection</td>
<td>1</td>
</tr>
<tr>
<td>MAX_AGE</td>
<td>Maximum age that foragers can reach</td>
<td>20 years</td>
</tr>
<tr>
<td>DEATH_RATE</td>
<td>Rate of stochastic death</td>
<td>0.1 per year</td>
</tr>
<tr>
<td>DIGESTION_TIME</td>
<td>Time interval at which digestion occurs</td>
<td>100 minutes</td>
</tr>
<tr>
<td>STOMACH_CAP</td>
<td>Number of items that can be eaten before forager is full</td>
<td>20</td>
</tr>
<tr>
<td>SEARCH_ANGLE</td>
<td>Angle about forward heading that the forager searches</td>
<td>0.1</td>
</tr>
<tr>
<td>SEARCH_DISTANCE</td>
<td>Maximum distance at which resource items can be detected</td>
<td>2 m</td>
</tr>
<tr>
<td>MOVE_SPEED</td>
<td>Speed when moving forward</td>
<td>3.6 km/h</td>
</tr>
<tr>
<td>STEP_DISTANCE</td>
<td>Distance of a single move forward event</td>
<td>1 m</td>
</tr>
<tr>
<td>EAT_TIME</td>
<td>Duration of EAT action</td>
<td>M</td>
</tr>
<tr>
<td>A</td>
<td>Area of the environment</td>
<td>39.2 km²</td>
</tr>
<tr>
<td>DENSITY</td>
<td>Total resource density</td>
<td>0.75</td>
</tr>
<tr>
<td>PATCH_ITEMS</td>
<td>Number of resource items in patch</td>
<td>12,800</td>
</tr>
<tr>
<td>PATCHES</td>
<td>Number of resource patches</td>
<td>24,500</td>
</tr>
<tr>
<td>Q_t</td>
<td>Maximum energy reward of resource type r</td>
<td>N(0.1, 0.1)</td>
</tr>
<tr>
<td>H_t</td>
<td>Practice time needed before obtaining half the maximal reward</td>
<td>0.1</td>
</tr>
<tr>
<td>S_r</td>
<td>Scalar for the sigmoid function describing how rewards increase with practice</td>
<td>1.4</td>
</tr>
<tr>
<td>Z</td>
<td>Noise on the reward foragers obtain from a resource; standard deviation of a normal distribution about ε_r</td>
<td>0.005</td>
</tr>
<tr>
<td>E_r</td>
<td>Rate at which resource species are replaced by new unknown species</td>
<td>0. R species per year</td>
</tr>
<tr>
<td>E_r</td>
<td>Rate at which resource species change their quality Q_r</td>
<td>0. R species per year</td>
</tr>
<tr>
<td>METABOLISM</td>
<td>Energy used for metabolism</td>
<td>0.001 per min</td>
</tr>
<tr>
<td>SATIATION_TIME</td>
<td>Time that satiation lasts</td>
<td>DIGESTION_TIME</td>
</tr>
<tr>
<td>SATIATION_FHAC</td>
<td>Proportion of stomach filled by a resource type in order to become satiated</td>
<td>0.9</td>
</tr>
<tr>
<td>G</td>
<td>Number of foragers in a group</td>
<td>20</td>
</tr>
<tr>
<td>SAFE_NEIGHS</td>
<td>Number of neighbors a forager needs to feel safe</td>
<td>9</td>
</tr>
<tr>
<td>SAFE_RANGE</td>
<td>SAFE_NEIGHS should be within this distance</td>
<td>17 m</td>
</tr>
<tr>
<td>PROB_CHECKSAFE</td>
<td>Probability to check if safe</td>
<td>0.9</td>
</tr>
<tr>
<td>ALIGN_PROB</td>
<td>Probability to align to neighs direction after MOVETOGROUP because not safe</td>
<td>0.9</td>
</tr>
<tr>
<td>ALIGN_ZONE</td>
<td>Zone of alignment</td>
<td>20 m</td>
</tr>
<tr>
<td>MOVETOGROUP_DISTANCE</td>
<td>Distance individual moves to become safe</td>
<td>3 m</td>
</tr>
<tr>
<td>MOVETOGROUP_SPEED</td>
<td>Speed at which forager move to become safe</td>
<td>180 m/min</td>
</tr>
<tr>
<td>SEARCH_TIME</td>
<td>Duration of SEARCH action</td>
<td>0.5 m</td>
</tr>
<tr>
<td>δ_t</td>
<td>Distance foragers move forward after not finding preferred food</td>
<td>0.20 m</td>
</tr>
<tr>
<td>φ_t</td>
<td>Selectivity scalar</td>
<td>0.20</td>
</tr>
<tr>
<td>γ_t</td>
<td>Rate scalar for updating the expected reward for the environment a_e</td>
<td>0.1</td>
</tr>
<tr>
<td>γ_t</td>
<td>Rate scalar for reinforcement learning</td>
<td>0.1</td>
</tr>
<tr>
<td>ν_t</td>
<td>Maximum exploration rate</td>
<td>0.1</td>
</tr>
<tr>
<td>T_t</td>
<td>Increment in probability to select a resource if previously stimulated in</td>
<td>0.1</td>
</tr>
<tr>
<td>ω_t</td>
<td>Probability to OBSERVE neighbor</td>
<td>0.1</td>
</tr>
<tr>
<td>τ_t</td>
<td>Maximum time observing neighbor</td>
<td>0.015 min</td>
</tr>
<tr>
<td>τ_t</td>
<td>Reward obtained by forager r from resource r</td>
<td>0. MAX_AGE</td>
</tr>
<tr>
<td>h_t</td>
<td>Forager’s total energy (or health)</td>
<td>0.8</td>
</tr>
<tr>
<td>r</td>
<td>Total time forager i has practiced processing resource r</td>
<td>0. MAX_AGE</td>
</tr>
<tr>
<td>a_k</td>
<td>Reward forager i expects from resource r</td>
<td>0. MAX_AGE</td>
</tr>
<tr>
<td>u_k</td>
<td>Reward forager i expects from environment</td>
<td>0. MAX_AGE</td>
</tr>
<tr>
<td>ε_t</td>
<td>Certainy estimate of expected reward</td>
<td>0.1</td>
</tr>
<tr>
<td>p_t</td>
<td>The time forager i has left to complete performing its present behavioral action</td>
<td>0. M</td>
</tr>
<tr>
<td>o_k</td>
<td>Effective observation time by forager i of neighbor k equals min(τ_e, p_k)</td>
<td>0. M</td>
</tr>
</tbody>
</table>

Table S1: List of parameters and variables. Upper case letters and names: to refer to invariant parameters that do not change during simulations. Lower case letters: variables of foragers that change during their lifetime. Greek letters: parameters that evolved. Subscripts: i = forager identity; r = resource type; k = forager’s neighbor; e = environment.
• Foraging patterns: larger-scale detection of resource patterns beyond the local perception of individuals (van der Post and Semmann, 2011a,b).
• Patterns of resource depletion (van der Post and Hogeweg, 2006).
• Interactions between grouping and resource patterns (van der Post and Hogeweg, 2006).
• Learning patterns (van der Post and Hogeweg, 2006).
• Cultural phenomena (van der Post and Hogeweg, 2008; van der Post and Hogeweg, 2009).

**Adaptation:** individuals have a decision making algorithm that determines how they respond to the local environment.

**Fitness:** we define that the rate of reproduction of foragers depends on energy intake, therefore more feeding generates a greater birth rate. A forager’s lifespan depends on a fixed death rate, starvation and maximum age.

**Prediction:** individuals do not predict future conditions, but respond to environmental conditions they encounter. These responses are embedded, in the sense that they are determined by locally observable conditions. We refer to this type of responsiveness as TODO, as in individuals DO WHAT THERE IS TO DO (Hogeweg and Hesper, 1985). What individuals do is determined by evolvable decision making algorithms.

**Sensing:** central for TODO is local information processing, leading to larger scale TODO-based pattern recognition beyond the direct perception of the individual (Hogeweg and Hesper, 1991; van der Post and Semmann, 2011a). We only define local perception of food (2 meters), and on a somewhat larger scale the awareness of neighbors.

**Interaction:** foragers can approach, eat and deplete resources. Foragers can detect each other and approach, align or move away from other individuals.

**Collectives:** resources patches and groups of foragers.

**Observation:** we collect data on what foragers eat, what their associations are, their experience, their group members, their age, energy, location, number of offspring, and evolvable parameters. Analysis generally proceeds via macro-observables (e.g. energy level) and becomes increasingly detailed as more information is needed to understand what and how processes are occurring (see main text).

1.5 Specification choices of scaling context

The scaling context of our model is determined by fixed parameters (Table S1):

**Spatial and temporal resolution:** At the minimal level we define the minimal time and spatial scale. Evolvable action durations can be minimally 10 seconds ($t_{MIN}$), to keep the model running fast enough. Space is continuous: individuals and resources can be placed at any given location.

**Local information processing:** Constraints on movement and perception (local information processing) are defined in terms of the spatial and temporal scale. Maximum speed is defined in terms of the minimal time interval, so that the duration of moving 1 meter can never be below $t_{MIN}$. The maximum range of resource detection ($r_D$) is then chosen to be small, so that individuals have to move to detect food (Video 1, gray semi-circle). Within this range all...
resource items are detected. The individual’s REACH $d_r$ is then chosen to be below maximal resource detection range (Video 1, gray small circle). This is quite a reasonable assumption (equal detection range and REACH is probably a limit case).

For grouping we assume that groups are cohesive and assume that forager’s can always keep track of all their group-mates, but ignore foragers of other groups. Moreover, for the alignment zone (ALIGN_ZONE) we assume a shorter range because alignment requires identifying the direction neighbors are moving.

Energy and life history: We fix the population at 100, and when a forager dies a parent is chosen according to its energy level $E_i$. The parent gives birth with cost $E_B$. This generates a viability threshold for the population, i.e. energy intake must be sufficient to have a net positive energy balance and must include sufficient energy for reproduction. If not, average energy levels drop and average energy levels in the population drop below $E_B$, in which case the population is not viable. In our simulations however, births and deaths continue.

Environment: The environmental settings were chosen to support selective foraging. This means that the resource density is so high that foragers do not end up eating everything, and need to be selective to reach the highest energy levels.

1.6 Initialization

We initialize the environment with a fixed number of resource types and multiple items of each type and patches with multiple resource types. An initial population of 100 foragers is initialized with parameters that have evolved. The population is initialized with 10 groups with 10 foragers each. Groups are initialized at random positions, where foragers start without an action, and with no information about the environment. Their first action will be FOODSCAN.

1.7 Input

The model does not include any external model or data files of driving environmental variables.

1.8 Submodels

Here we present a “mathematical” skeleton of the model updating schedule:

1.8.1 Forager continuous updates and decision making algorithm

During each minute:
while any action ends this minute, take soonest ending action:
1) Complete Action
   —-* MOVE: get new position
   —-* FOODSCAN: get new food target, or not
   —-* MOVETOFOOD: get new position
   —-* MOVETOGROUP: get new position
   —-* EAT: gain energy $E_i = E_i + e_{ir}$ and gain experience: $t_{ir}$ and $a_{ir}$
     where $E_i$ is individual energy and $e_{ir}$ is energy per food item
   —-* OBSERVE: gain experience $t_{ir}$
—* NOTHING: ready for next action

2) Choose New Action based on previous action:
   —→ if was MOVE: reselect MOVE if not yet reached $\delta_i$ meters
   —→ if NO ACTION YET ASSIGNED
   —→* CHECK IF SAFE and update any stimulus enhancement
   —→* if (not safe) then MOVETOGROUP
   —→* else (safe)
   —→** if was MOVETOGROUP then align to the direction of group members in ALIGN_ZONE
   —→** if was SEARCH or MOVETOFOOD then
   —→*** if resource target still exists
   —→**** if (distance to target < $d_R$)
   —→***** EAT
   —→**** else
   —→***** MOVETOFOOD
   —→*** give up (food is gone)
   —→** if no action assigned and (RAND < $\omega_i$) then OBSERVE
   —→** else (no action and not observe)
   —→**** do NOTHING
   —→*** else
   —→**** if (previous action != FOODSCAN) and not crowded
   —→***** FOODSCAN
   —→**** else
   —→***** MOVE

3) Get new completion time

4) Place forager in action cue sorted by completion time.

RAND is a random number between 0 and 1, $\omega_i$ is the probability to OBSERVE, and $d_R$ is the individual’s reach.

1.8.2 Behavior actions of foragers

MOVE:

- Get new position and make sure it is on the grid:
  
  \[
  \vec{p}' = d_M \vec{v} + \vec{p}
  \]
  
  where $d_M$ is the MOVE distance, $\vec{v}$ = normalized random vector, and $\vec{p}$ is the position of the forager.

- while $(p'_x < 0$ or $p'_y < 0$ or $p'_x > 0$ or $p'_y > \sqrt{A})$

  * select new $\vec{v}$ (normalized random vector)
  * $\vec{p}' = d_M \vec{v} + \vec{p}$

  where $A$ is the area of the environment.
• Duration: \( d_M \times t_M \)
  where \( t_M \) is the time it takes the forager to move 1 meter.

MOVETOGROUP:
  \[ \vec{v} = \sum_{j \neq i}^{n_A} \vec{p}_j \]
  \( n_A \) is the number of group members and \( \vec{p}_j \) is the position of neighbor \( j \).

  \( \vec{p}' = d_M \vec{v} + \vec{p} \)
  where \( d_M \) is the MOVE distance and \( \vec{p} \) is the position of the forager.

• Duration: \( d_M \times t_M \)
  where \( t_M \) is the time it takes the forager to move 1 meter, \( d_M \) is the step-distance.

FOODSCAN (Video 1, gray semi-circle): try to get new food target
  • for all \( n \) food items (distance < \( d_F \) and angle < \( a_F/2 \) from heading)
    where \( d_F \) is the radius of search, and \( a_F \) is the search angle about the foragers forward heading.
    – assess a maximum of 20 items
    – for each item decide whether to eat of not based on Equation 1.

• Duration: \( t_F \)

MOVETOFOOD: get new position closer to food item
  \[ \vec{v} = \frac{\vec{f} - \vec{p}}{|\vec{f} - \vec{p}|} \]
  \( \vec{d} = \) distance to food where \( \vec{f} \) is the position of food item and \( \vec{p} \) the position of the forager

  • if \( d > d_r \)
    \[ \vec{p}' = d_M \vec{v} + \vec{p} \]
  else
    \[ \vec{p}' = d\vec{v} + \vec{p} \]

• Duration: distance moved \( \times t_M \)

EAT: consume food item
• Duration: \( t_E \)

OBSERVE: observe neighbor
• Duration: \( \tau_i \)
  • leads to acquisition of experience based on Equation 8.

NOTHING: do nothing
• Duration: \( t_N \)
1.8.3 Forager discrete updates

Each minute for all foragers:

- Age: age + 1 minute
- Energy: $E_i = E_i - E_m$
  where $E_m$ is energy metabolism per minute
- Stimulus enhancement time (after stimulus):
  - if (stimulus duration < 30 minutes): stimulus time = stimulus time + 1 minute
  - if (stimulus duration \( \geq \) 30) no more stimulus
- Reproduction:
  - if (population < 100): select parent according to energy (see Equation S1)
    * Energy change due to birth: $E_i = E_i - E_B$, where $E_B$ is the birth cost.
    * Offspring energy: $E_B$
- Death (forager dies if):
  - $E_i \leq 0$
  - age \( \geq \) maximum age
  - RAND < death rate (equivalent to 0.1 per year)

1.8.4 Resources

For all depleted resource items every year:

- all depleted resource items re-appear (or re-grow).
- resource items that are eaten disappear immediately.

This is not computed by looping over all items all the time, but only when those items are assessed by foragers locally.

1.9 Comparison to diet model by van der Post & Hogeweg

The present model differs from that of van der Post and Hogeweg (2006) in the following ways:
(i) Here foragers can immediately sense rewards, while in van der Post and Hogeweg (2006) this was delayed to after digestion, enabling sensing of rewards to represent a mixture of resources. The latter makes learning complicated. This is not considered here in order to focus on skill learning, but in principle both skill learning and mixing of rewards could occur at the same time.
(ii) In the present study we add skill learning and rewards increase of time. In van der Post and Hogeweg (2006) rewards were constant, but assessments thereof would develop from zero.
(iii) To enable foragers to try unknown and less known resources, we implement an intrinsic
exploration tendency that depends on the forager’s certainty estimate. In van der Post and Hogeweg (2006) this was implicit by assuming that individuals only explore resources for which they have no existing preference or association with a probability depending on ‘trail rate’. The latter could effectively limit exploration via a familiarity-selectivity feedback. This could generate unfair implicit constraints on asocial learning, relative to which an advantage of social learning would be an artifact. Hence we added a more general exploration tendency, mitigated by knowledge of a resource (‘certainty’, see Equation 2).
2 Reduction of traditional differences with increasing task difficulty

Here we explain why the magnitude of traditional differences declines as task difficulty increases beyond H=1 (Fig. 2A). This pattern appears to be explained by a similar pattern of between-group differences declining once $H > 1$ (Fig. 3A). We therefore need to explain why between-group differences decline once $H > 1$.

Between-group differences arise when there are limits on population-wide convergence on particular subsets of resource types. For a given simulation, resources in our model only vary with respect to two characteristics: (i) resource quality $Q_r$ and (ii) the rate at which rewards increase with practice defined by $S_r$ (Equation 4, Fig 1C). Population-wide convergence on high quality resources is expected through the process of optimization. Convergence on particular values of $S_r$ was not expected a priori, but is a possibility.

Our results show that as task difficulty increases, optimization expressed in terms of repertoire quality, becomes more limited (Fig 3C). As a result we would expect the scope for between-group differences to increase as $H$ increases, since population-wide convergence on an optimum was becoming more limited. This implies that convergence on high quality resources is not the cause for a decline in between-group differences when $H > 1$. Thus we focus on $S_r$.

In Fig. S1 we show the mean value of $S_r$ of consumed resources for different levels of task difficulty ($H$). Any convergence on extreme values of $S_r$ will be revealed if the mean of $S_r$ if either high (close to 4) or low (close to 1). The mean $S_r$ value is indicated by the horizontal
gray line. We can observe that as $H$ increases, the mean value of $S_r$ declines and approaches one, implying a population-wide convergence on resource types with low $S_r$.

Low $S_r$ means that for the same quality resource and practice time, foragers obtain a greater reward, than for resource types with high $S_r$. As a result, there is a stronger feedback between learning and preference, and stronger familiarity biases towards resources with low $S_r$. Thus as task difficulty increases, there is an increasing bias to the subset of resources that is easiest to learn.

References


