

Supplementary material to the paper “Multi-armed bandit for species discovery: a Bayesian nonparametric approach”

Marco Battiston, Stefano Favaro and Yee Whye Teh

1. PROOFS.

In the following proofs, we assume all random variables to be defined on a common probability space, and we denote by \mathbb{P} its probability measure and by \mathbb{E} the corresponding expectation operator. Furthermore, before proving Proposition 1 and Proposition 2, we recall the following result by [Gnedin and Pitman \(2006\)](#) that will be used in our proofs.

Proposition *Let $P \sim PY(\sigma, \theta, H)$ and let (Y_1, \dots, Y_n) be a sample from it. Then, the probability of observing k distinct values in (Y_1, \dots, Y_n) is denoted by $F(n, k, \sigma, \theta)$ and*

$$F(n, k, \sigma, \theta) = \frac{\prod_{r=1}^{k-1} (\theta + r\sigma)}{\sigma^k (\theta + 1)_{n-1}} \mathcal{C}(n, k, \sigma)$$

where \mathcal{C} is the generalized factorial coefficient, defined for all $n \in \mathbb{N}, k \leq n, 0 \leq \sigma \leq 1$ as $\mathcal{C}(n, k; \sigma) = (1/k!) \cdot \sum_{0 \leq j \leq k} (-1)^j \binom{k}{j} (-j\sigma)_n$, with the proviso $\mathcal{C}(0, 0; \sigma) = 1$ and $\mathcal{C}(n, 0; \sigma) = 0 \ \forall n$ and where $(\theta + 1)_{n-1} = (\theta + 1)(\theta + 2) \cdots (\theta + n - 1)$ is the rising factorial coefficient.

We also recall the characterization of the posterior distribution of the PY process derived in [Pitman \(1996\)](#), and the Chinese Restaurant Representation of the HPY.

Given a sample $\mathbf{Y}_n = (Y_1, \dots, Y_n)$, such that $Y_i|P \stackrel{iid}{\sim} P$ for all $1 \leq i \leq n$ and $P \sim \text{PY}(\sigma, \theta, H)$, the posterior of P given \mathbf{Y}_n satisfies the following distributional equation

$$P|\mathbf{Y}_n \stackrel{d}{=} \sum_{i=1}^{K_n} w_i \delta_{Y_i^*} + w_0 \tilde{P} \quad (1)$$

where K_n is the number of distinct values in the sample \mathbf{Y}_n , denoted by $(Y_1^*, \dots, Y_{K_n}^*)$ and having multiplicities (n_1, \dots, n_{K_n}) , $(w_0, w_1, \dots, w_{K_n})$ is a random vector distributed according to $\text{Dir}(\theta + K_n\sigma, n_1 - \sigma, \dots, n_{K_n} - \sigma)$ and $\tilde{P} \sim \text{PY}(\sigma, \theta + K_n\sigma, H)$.

The Chinese Restaurant Franchise representation of the HPY process is described by the following two predictive distributions for the observables and for the cluster values in population j

$$Y_{j,i+1}|Y_{j,1}, \dots, Y_{j,i}, \sigma_j, \theta_j, P_0 \sim \sum_{t=1}^{m_{j\cdot}} \frac{n_{jt\cdot} - \sigma_j}{\theta_j + n_{j\cdot}} \delta_{Y_{j,t}^*} + \frac{\theta_j + m_{j\cdot}\sigma_j}{\theta_j + n_{j\cdot}} P_0 \quad (2)$$

and

$$Y_{j,m_{j\cdot}+1}^*|Y_{1,1}^*, \dots, Y_{J,m_J}^*, \alpha, \gamma, H \sim \sum_{k=1}^K \frac{m_{\cdot k} - \alpha}{\gamma + m_{\cdot\cdot}} \delta_{Y_k^{**}} + \frac{\gamma + K\alpha}{\gamma + m_{\cdot\cdot}} H. \quad (3)$$

Proof of Proposition 1. From Equation (3), the franchise-wide distinct values $(Y_1^{**}, \dots, Y_K^{**})$ are governed by P_0 and $P_0 \sim \text{PY}(\alpha, \gamma, H)$. Using formula (1), the posterior distribution of P_0 , given the observations, satisfies the distributional equation

$$P_0|\mathbf{Y}_n \stackrel{d}{=} \sum_{k=1}^K \beta_k \delta_{Y_k^{**}} + \beta_0 P_0'$$

where

$$P_0'|\mathbf{Y}_n \sim \text{PY}(\alpha, \gamma + K\alpha, H)$$

$$\beta|\mathbf{Y}_n = (\beta_0, \dots, \beta_K)|\mathbf{Y}_n \sim \text{Dir}(\gamma + K\alpha, m_{\cdot 1} - \alpha, \dots, m_{\cdot K} - \alpha)$$

Similarly, from formula (2), we can apply formula (1) to P_j to find a distributional equation for P_j , conditionally on P_0 and the data. Also, using the distributional equation for the posterior of P_0 , we find the following distributional equation for P_j

$$P_j | \beta, P'_0, \mathbf{Y}_n \stackrel{d}{=} \sum_{k=1}^K \pi_{j,k} \delta_{Y_k^{**}} + \pi_{j,0} P'_j \quad (4)$$

where

$$\begin{aligned} P'_j | P'_0, \mathbf{Y}_n &\sim \text{PY}(\sigma_j, (\theta_j + m_{j \cdot} \sigma_j) \beta_0, P'_0) \\ (\pi_{j,0}, \dots, \pi_{j,K}) | \beta, \mathbf{Y}_n &\sim \text{Dir}((\theta_j + m_{j \cdot} \sigma_j) \beta_0, (\theta_j + m_{j \cdot} \sigma_j) \beta_1 + n_{j \cdot 1} - \sigma_j m_{j1}, \dots \\ &\quad \dots, (\theta_j + m_{j \cdot} \sigma_j) \beta_K + n_{j \cdot K} - \sigma_j m_{jK}) \end{aligned}$$

So, the distribution of $P_j(A) | \mathbf{Y}_n, P_0$ satisfies

$$P_j(A) | \beta, P'_0, \mathbf{Y}_n \stackrel{d}{=} \sum_{k=1}^K \pi_{j,k} \delta_{Y_k^{**}}(A) + \pi_{j,0} P'_j(A)$$

for all $j \in \{1, \dots, J\}$, which implies

$$P_j(A) | \beta_0, \mathbf{Y}_n \sim \text{beta}((\theta_j + m_{j \cdot} \sigma_j) \beta_0, (\theta_j + m_{j \cdot} \sigma_j) (1 - \beta_0) + n_{j \cdot} - \sigma_j m_{j \cdot})$$

where we made use of the following facts:

1. $\delta_{Y_k^{**}}(A) = 0 \ \forall k = 1, \dots, K$: since $\{Y_1^{**}, \dots, Y_K^{**}\} = A^c$.
2. $P'_j(A) \stackrel{as}{=} 1$: P'_j can be rewritten as $P'_j = \sum_{i \geq 1} \gamma_i \delta_{X_i}$ for some weights $\{\gamma_i\}_{i \geq 1}$ and atoms $\{X_i\}_{i \geq 1} \stackrel{iid}{\sim} H$. Then, $\mathbb{P}(\cap_{i \geq 1} \{X_i \in A\}) = \prod_{i \geq 1} \mathbb{P}(X_i \in A) = \prod_{i \geq 1} 1 = 1$, since H is diffuse and A^c is a finite set of points. Finally, $\mathbb{P}(\cap_{i \geq 1} \{X_i \in A\}) = 1 \Rightarrow P'_j(A) \stackrel{as}{=} 1$.
3. $\pi_{j,0} | \beta_0, \mathbf{Y}_n \sim \text{beta}((\theta_j + m_{j \cdot} \sigma_j) \beta_0, (\theta_j + m_{j \cdot} \sigma_j) (1 - \beta_0) + n_{j \cdot} - \sigma_j m_{j \cdot})$: by the aggregation property of Dirichlet distribution.

Also, since we are conditioning on P_0 (through β, P'_0), $P_j(A) | \beta_0, \mathbf{Y}_n$ is independent of $P_i(A) | \beta_0, \mathbf{Y}_n$ for all $i, j \in \{1, \dots, J\}$, $i \neq j$. Hence, their joint distribution is simply the product of the marginals. The last step is to integrate β_0 out

$$(P_1(A), \dots, P_J(A)) | \mathbf{Y}_n = \int_0^1 \prod_{j=1}^J P_j(A) | \beta_0, \mathbf{Y}_n \cdot dF_{\beta_0}(\beta_0)$$

where the distribution of β_0 is another beta (again by aggregation of Dirichlet distribution). So, $(P_1(A), \dots, P_J(A)) | \mathbf{Y}_n$ admits a density as stated.

Proof of Proposition 2. Using the distributional equation (4) for the posterior of P_j and working conditionally on $\beta_0 | \mathbf{Y}_n \sim \text{beta}(\gamma + K\alpha, m_{..} - \alpha K)$, we compute $\mathbb{P}(K_j^{(z)} = k | \mathbf{Y}_n, \beta_0)$.

From the distributional equation, we know that, given

$$\pi_{j,0} | \beta_0, \mathbf{Y}_n \sim \text{beta}((\theta_j + m_{j \cdot} \sigma_j) \beta_0, (\theta_j + m_{j \cdot} \sigma_j) (1 - \beta_0) + n_{j \cdot} - \sigma_j m_{j \cdot})$$

an observation $Y_{n_{j \cdot} + i}$ with $i = 1, \dots, z$ does not coincide with any of the K distinct species (in the joint sample) with probability $\pi_{j,0}$. To have $K_j^{(z)} = k$, at least k of the z data $Y_{n_{j \cdot} + 1}, \dots, Y_{n_{j \cdot} + z}$ must be allocated to the k new distinct species that have not previously observed. Hence,

$$\mathbb{P}(K_j^{(z)} = k | \mathbf{Y}_n, \beta_0, \pi_{j,0}) = \sum_{i=k}^z \binom{z}{i} \pi_{j,0}^i (1 - \pi_{j,0})^{z-i} \mathbb{P}(K_i = k | \beta_0)$$

where K_i is now the number of distinct species in a sample of size i generated by a $\text{PY}(\sigma_j, (\theta_j + m_{j \cdot} \sigma_j) \beta_0, P'_0)$, where $P'_0 \sim \text{PY}(\alpha, \gamma + K\alpha, H)$.

We need to find $\mathbb{P}(K_i = k | \beta_0)$. Using the Chinese Franchise Representation and the result by [Gnedin and Pitman \(2006\)](#), denoting by M_i the number of tables, we have that, for $\tilde{m} = 1, \dots, i$, $\mathbb{P}(M_i = \tilde{m}) = F(i, \tilde{m}, \sigma_j, (\theta_j + m_{j \cdot} \sigma_j) \beta_0)$. Moreover, conditionally on $M_i = \tilde{m}$, for $k = 1, \dots, \tilde{m}$, $\mathbb{P}(K_i = k | M_i = \tilde{m}) = F(\tilde{m}, k, \alpha, \gamma + K\alpha)$. Finally, $\mathbb{P}(K_j^{(z)} = k | \mathbf{Y}_n, \beta_0, \pi_{j,0})$ can be computed as

$$\sum_{i=k}^z \binom{z}{i} \pi_{j,0}^i (1 - \pi_{j,0})^{z-i} \sum_{\tilde{m}=k}^i F(\tilde{m}, k, \alpha, \gamma + K\alpha) F(i, \tilde{m}, \sigma_j, (\theta_j + m_{j \cdot} \sigma_j) \beta_0)$$

The conditional mean $\mathbb{E}(K_j^{(z)} | \mathbf{Y}_n, \beta_0, \pi_{j,0})$ is found by averaging over $\{0, \dots, z\}$ and, being constant, they are trivially independent among arms. Hence, the joint distribution of $(\mathbb{E}(K_1^{(z)} | \mathbf{Y}_n), \dots, \mathbb{E}(K_J^{(z)} | \mathbf{Y}_n))$ is found by integrating $\beta_0, (\pi_{j,0} : j \in \{1, \dots, J\})$ out from the product of these J conditional (constant) distributions.

2. IMPLEMENTATION ISSUES - MCMC ALGORITHM FOR THE HPY PARAMETERS.

The number of clusters in each population $\mathbf{m}_J = (m_{j\cdot} : j \in \{1, \dots, J\})$ appearing in the parametrization of the beta distributions in both Algorithms 1 and 2 of Section 3.1 and 3.2 of the paper, are latent variables. In subsection 2.1 we describe a simple MCMC scheme to estimate them in case an initial sample is available. The MCMC algorithm directly follows from paragraph 5.1 of [Teh et al. \(2006\)](#). Moreover if the hyperparameters of the HPY model are unknown, they must added to the MCMC sampler too, as outlined in subsection 2.2.

2.1. MCMC for \mathbf{m}_J : In principle a Gibbs sampler to estimate \mathbf{m}_J should sequentially draw samples from the full conditionals $\pi(m_{j\cdot} | m_{1\cdot}, \dots, m_{j-1\cdot}, m_{j+1\cdot}, \dots, m_{J\cdot}, \mathbf{Y}_n)$. However both the joint $\pi(m_{1\cdot}, \dots, m_{J\cdot} | \mathbf{Y}_n)$ and the full conditional posterior distributions are difficult combinatorial objects and cannot be derived in closed form. A possible solution is a Gibbs sampler that, rather than directly updating $m_{j\cdot} | m_{-j\cdot}, \mathbf{Y}_n$, updates the cluster allocations $(t_{ji} : i \in \{1, \dots, n_{j\cdot}\})$ and then computes $m_{j\cdot} | m_{-j\cdot}, \mathbf{Y}_n$. As in [Teh et al. \(2006\)](#), the cluster allocation variable t_{ji} specifies the cluster to which the i -th observation of population j belongs. Let $\mathbf{t}_{-jp}^{(i-1)}$ denote the array of cluster allocations after iteration $i-1$ of the sampler and with the p -th observation of the j -th population removed. Then $t_{jp}^{(i)} | (\mathbf{Y}_n, \mathbf{t}_{-jp}^{(i-1)})$ is proportional to

$$\sum_{t: \psi_{jt} = \psi_{jt_{ji}}} \frac{n_{jt\cdot} - I\left(t = t_{jp}^{(i-1)}\right) - \sigma_j}{\theta_j + n_{j\cdot} - 1} \delta_t + \frac{\theta_j + m_{j\cdot}^{(i-1, p-1)} \sigma_j}{\theta_j + n_{j\cdot} - 1} \frac{m_{\cdot k_{jp}}^{(i-1, p-1)} - \alpha}{\gamma + m_{\cdot\cdot}^{(i-1, p-1)}} \delta\left(m_{j\cdot}^{(i-1, p-1)} + 1\right)$$

where $m_{j\cdot}^{(i-1, p-1)}$ denotes the number of clusters in population j at the i -th iteration after having updated the first $p-1$ cluster allocations of that population, ψ_{jt} is a classification variable that tells us the species of the observations in the t -th cluster in population j and k_{jp} is the species of the observations in the p -th cluster in population

j . If $n_{jt_{jp}} = 1$ (i.e. the observation is forming its own cluster), before updating $t_{jp}^{(i)}$ we must remove its cluster and subtract one to all the m 's. The updated value for $m_j^{(i)}$ can also be taken as the highest $t_{jp}^{(i)}$ for $p \in \{1, \dots, n_{j..}\}$, rather than the number of distinct values in the $t_{jp}^{(i)}$.

The algorithm is time expensive because at every iteration it re-samples the cluster allocations of all populations and of all observations. However, we experienced that a good choice of the starting value makes the chain converge to its stationary distribution in just a few iterations. We suggest to run a Chinese Franchise given the data to find the initial point for cluster allocations to start the Gibbs sampler.

When the HPY-TS algorithm is run, the vector \mathbf{m}_J can be updated by allocating new observations to either old or new clusters using the Chinese Restaurant Franchise. If the observation is new, it forms a new cluster. If it is old, say of type Y_k^{**} , then the corresponding observation either will form a new cluster with probability proportional to $((m_{..k} - \alpha)/(\gamma + m_{..}))(\theta_j + m_{j..}\sigma_j)/((\theta_j + n_{j..}))$ or it will join an existing cluster (with dish Y_k^{**}) with probability proportional to $(n_{j..k} - m_{jk}\sigma_j)/(\theta_j + n_{j..})$.

2.2. HPY Hyperparameters: If the hyperparameters are considered as unknown, they must be included in the Gibbs sampler for the cluster sizes. Assuming independent priors for hyperparameters of different Pitman-Yor processes, the full conditional distributions can be derived from

$$\begin{aligned} \pi(\alpha, \gamma | (m_{jk} : j \in \{1, \dots, J\}, k \in \{1, \dots, K\}), (\sigma_j, \theta_j : j \in \{1, \dots, J\}), \mathbf{Y}_n) = \\ = \pi(\alpha, \gamma | m_{..}, K) \propto \frac{\Gamma(\frac{\gamma}{\alpha} + K)\Gamma(\gamma)\mathcal{C}(m_{..}, K, \alpha)}{\Gamma(\frac{\gamma}{\alpha})\Gamma(\gamma + m_{..})}\pi^{prior}(\alpha, \gamma) \end{aligned}$$

and, for each couple $((\sigma_j, \theta_j) : j \in \{1, \dots, J\})$, from

$$\begin{aligned} \pi(\sigma_j, \theta_j | (m_{jk} : j \in \{1, \dots, J\}, k \in \{1, \dots, K\}), \sigma_{-j}, \theta_{-j}, \alpha, \gamma, \mathbf{Y}_n) = \\ = \pi(\sigma_j, \theta_j | n_{j..}, m_{j..}) \propto \frac{\Gamma(\frac{\theta_j}{\sigma_j} + m_{j..})\Gamma(\theta_j)\mathcal{C}(n_{j..}, m_{j..}, \sigma_j)}{\Gamma(\frac{\theta_j}{\sigma_j})\Gamma(\theta_j + n_{j..})}\pi^{prior}(\sigma_j, \theta_j) \end{aligned}$$

3. SIMULATIONS RESULTS - TABLES OF WEIGHTS.

The following tables report the weights given to each arm by the four algorithms in the simulation study. We consider the behaviour of the HPY-TS algorithm and of the three other competing strategies in the 3 scenarios described in Section 4.1 of the paper. Specifically, we consider the following three scenarios:

1. *Pure Exploitation Scenario*, corresponding to the Zipf parameter vector for the true distributions equal to $(1.3, 1.3, 2, 2, 2, 2, 2, 2)$;
2. *Pure Exploration Scenario*, corresponding to the Zipf parameter vector $(1.3, 1.3, 1.3, 1.3, 1.3, 1.3, 2, 2)$;
3. *Exploration-Exploitation Scenario*, corresponding to the Zipf parameter vector $(1.3, 1.3, 1.3, 1.3, 2, 2, 2, 2)$.

For each scenario we run the four algorithms for 60 times. We report here only the results of the first 10 simulations. Each row in the tables is the result of one simulation. The columns correspond to the possible arms. Finally, we repeat the same simulations both for abundance and for incidence data.

References

- Gnedin, A. and Pitman, J. (2006). Exchangeable Gibbs partitions and Stirling triangles. *Journal of Mathematical Science*, **138**, 5674–5685.
- Pitman, J. (1996). Some developments of the Blackwell-MacQueen urn scheme. In *Statistics, Probability and Game Theory* (eds. T. S. Ferguson, L. S. Shapley and J. B. MacQueen), 245–267. Institute of Mathematical Statistics, Hayward.
- Teh, Y.W., Jordan, M., Beal, M. and Blei, D. (2006). Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, **101**, 1566–1581.

Table 1: Simulations: Pure Exploitation. Abundance data.

	HPY-TS								Good-Turing							
Runs/Zipf	1.3	1.3	2	2	2	2	2	2	1.3	1.3	2	2	2	2	2	2
1	157	137	4	1	1	0	0	0	3	292	1	1	1	1	1	0
2	147	121	3	5	1	4	13	6	282	11	1	1	1	1	2	1
3	168	125	0	0	0	4	2	1	2	298	0	0	0	0	0	0
4	10	279	2	0	6	0	2	1	5	269	4	4	4	5	4	5
5	216	66	0	0	5	3	0	10	293	1	1	1	1	1	1	1
6	134	150	0	1	0	12	0	3	1	298	0	0	0	1	0	0
7	147	100	1	9	1	35	0	7	291	2	1	1	1	2	1	1
8	137	161	0	2	0	0	0	0	290	5	1	1	0	1	1	1
9	146	149	1	0	0	4	0	0	299	1	0	0	0	0	0	0
10	134	111	8	3	4	6	13	21	288	5	1	1	1	1	1	2
	Uniform								Oracle							
Runs/Zipf	1.3	1.3	2	2	2	2	2	2	1.3	1.3	2	2	2	2	2	2
1	31	48	34	39	40	32	38	38	139	161	0	0	0	0	0	0
2	45	39	33	43	38	26	39	37	178	122	0	0	0	0	0	0
3	40	39	41	33	41	31	34	41	147	153	0	0	0	0	0	0
4	51	35	40	28	32	27	48	39	157	143	0	0	0	0	0	0
5	40	40	41	36	29	42	30	42	112	188	0	0	0	0	0	0
6	26	45	37	42	39	37	36	38	176	124	0	0	0	0	0	0
7	38	38	56	37	31	29	30	41	126	174	0	0	0	0	0	0
8	29	38	38	40	43	31	47	34	197	103	0	0	0	0	0	0
9	39	30	37	37	47	43	35	32	179	121	0	0	0	0	0	0
10	36	31	34	38	31	39	46	45	120	180	0	0	0	0	0	0

Table 2: Simulations: Pure Exploitation. Incidence data.

	HPY-TS								Good-Turing							
Runs/Zipf	1.3	1.3	2	2	2	2	2	2	1.3	1.3	2	2	2	2	2	2
1	14	16	0	0	0	0	0	0	5	9	2	2	3	3	3	3
2	15	14	0	0	0	1	0	0	3	15	1	2	2	2	2	3
3	22	6	0	0	0	0	2	0	7	6	3	3	2	3	3	3
4	9	19	0	2	0	0	0	0	24	1	1	1	1	0	1	1
5	11	19	0	0	0	0	0	0	14	3	2	2	3	2	2	2
6	10	20	0	0	0	0	0	0	14	4	1	2	3	2	2	2
7	16	13	0	0	0	0	0	1	4	13	2	2	2	3	2	2
8	17	12	0	1	0	0	0	0	6	11	3	2	2	2	2	2
9	16	14	0	0	0	0	0	0	18	2	2	2	1	2	1	2
10	13	16	0	0	0	0	1	0	11	5	2	3	2	2	2	3
	Uniform								Oracle							
Runs/Zipf	1.3	1.3	2	2	2	2	2	2	1.3	1.3	2	2	2	2	2	2
1	4	5	6	3	3	0	7	2	15	15	0	0	0	0	0	0
2	4	4	3	3	2	6	4	4	15	15	0	0	0	0	0	0
3	2	4	6	4	3	5	2	4	13	17	0	0	0	0	0	0
4	4	8	3	2	3	2	4	4	20	10	0	0	0	0	0	0
5	5	2	1	3	5	3	6	5	12	18	0	0	0	0	0	0
6	3	10	1	2	4	4	3	3	12	17	1	0	0	0	0	0
7	3	2	2	4	3	4	6	6	15	15	0	0	0	0	0	0
8	7	5	4	0	2	2	4	6	12	18	0	0	0	0	0	0
9	2	2	9	1	2	5	3	6	13	17	0	0	0	0	0	0
10	2	6	4	2	7	5	2	2	16	14	0	0	0	0	0	0

Table 3: Simulations: Pure Exploration. Abundance data.

	HPY-TS								Good-Turing							
Runs/Zipf	1.3	1.3	1.3	1.3	1.3	1.3	2	2	1.3	1.3	1.3	1.3	1.3	1.3	2	2
1	43	47	41	68	30	71	0	0	2	4	1	1	1	290	1	0
2	1	61	37	15	100	86	0	0	0	2	295	1	1	1	0	0
3	11	40	148	57	40	3	0	1	2	287	4	2	2	1	1	1
4	46	114	15	47	64	14	0	0	298	0	0	0	2	0	0	0
5	4	112	1	43	37	103	0	0	2	286	1	3	2	5	0	1
6	56	47	48	21	100	28	0	0	2	2	289	1	5	1	0	0
7	42	53	9	67	98	31	0	0	6	2	1	3	286	2	0	0
8	58	25	31	51	34	101	0	0	285	2	7	3	1	1	1	0
9	36	68	107	37	1	51	0	0	1	3	293	1	1	1	0	0
10	36	35	44	68	41	76	0	0	2	2	1	2	280	12	1	0
	Uniform								Oracle							
Runs/Zipf	1.3	1.3	1.3	1.3	1.3	1.3	2	2	1.3	1.3	1.3	1.3	1.3	1.3	2	2
1	31	43	31	34	38	49	30	44	83	41	30	58	54	34	0	0
2	45	24	29	36	32	40	50	44	73	41	74	77	5	30	0	0
3	43	47	32	36	38	28	35	41	38	31	61	78	48	44	0	0
4	27	44	37	45	41	30	41	35	46	30	60	42	54	68	0	0
5	33	40	49	44	34	30	35	35	52	45	42	49	35	77	0	0
6	39	36	42	40	33	36	33	41	75	57	75	36	10	47	0	0
7	31	36	43	54	30	42	30	34	38	60	108	19	51	24	0	0
8	49	32	31	36	39	42	35	36	37	72	19	86	53	33	0	0
9	33	39	42	36	36	40	34	40	53	52	51	17	69	58	0	0
10	40	33	30	33	40	45	39	40	55	73	47	36	61	28	0	0

Table 4: Simulations: Pure Exploration. Incidence data.

	HPY-TS								Good-Turing							
Runs/Zipf	1.3	1.3	1.3	1.3	1.3	1.3	2	2	1.3	1.3	1.3	1.3	1.3	1.3	2	2
1	0	14	5	6	2	3	0	0	2	2	5	15	2	2	1	1
2	6	8	5	5	6	0	0	0	2	5	15	2	2	2	1	1
3	1	3	4	7	7	8	0	0	3	1	19	1	1	3	1	1
4	7	4	0	5	4	10	0	0	5	2	2	2	8	9	1	1
5	6	8	7	7	2	0	0	0	2	7	6	7	2	3	2	1
6	1	16	2	5	2	4	0	0	1	1	1	25	1	1	0	0
7	10	1	6	1	9	3	0	0	1	1	24	1	1	2	0	0
8	5	12	1	2	2	8	0	0	3	1	1	1	2	21	0	1
9	7	5	10	2	0	6	0	0	4	4	2	1	1	16	1	1
10	3	0	15	0	6	6	0	0	1	1	3	3	18	2	1	1
	Uniform								Oracle							
Runs/Zipf	1.3	1.3	1.3	1.3	1.3	1.3	2	2	1.3	1.3	1.3	1.3	1.3	1.3	2	2
1	4	4	3	5	4	3	3	4	6	5	1	6	7	5	0	0
2	3	4	4	3	6	4	3	3	5	8	4	4	4	5	0	0
3	4	4	3	5	3	5	2	4	6	10	3	7	2	2	0	0
4	7	0	4	3	6	5	2	3	4	2	5	6	9	4	0	0
5	3	5	3	2	3	6	3	5	3	4	4	3	10	6	0	0
6	4	6	3	3	8	2	1	3	5	5	5	5	6	4	0	0
7	3	3	1	3	5	7	6	2	8	4	3	3	4	8	0	0
8	3	7	5	2	4	3	1	5	3	4	3	6	6	8	0	0
9	3	4	2	7	5	4	2	3	3	7	6	4	6	4	0	0
10	6	3	2	3	6	2	6	2	5	5	7	4	4	5	0	0

Table 5: Simulations: Exploration-Exploitation. Abundance data.

	HPY-TS								Good-Turing							
Runs/Zipf	1.3	1.3	1.3	1.3	2	2	2	2	1.3	1.3	1.3	1.3	2	2	2	2
1	10	159	74	57	0	0	0	0	0	0	300	0	0	0	0	0
2	81	76	63	80	0	0	0	0	297	1	1	1	0	0	0	0
3	72	60	85	83	0	0	0	0	1	2	4	292	1	0	0	0
4	72	30	129	69	0	0	0	0	1	1	2	294	1	0	0	1
5	107	42	59	90	0	0	2	0	298	0	1	1	0	0	0	0
6	63	105	112	20	0	0	0	0	298	2	0	0	0	0	0	0
7	92	86	42	78	0	1	1	0	4	2	3	290	0	0	1	0
8	79	120	27	73	0	0	1	0	293	5	1	1	0	0	0	0
9	61	49	108	81	0	0	0	1	296	1	1	2	0	0	0	0
10	100	35	57	108	0	0	0	0	300	0	0	0	0	0	0	0
	Uniform								Oracle							
Runs/Zipf	1.3	1.3	1.3	1.3	2	2	2	2	1.3	1.3	1.3	1.3	2	2	2	2
1	37	35	36	54	36	29	38	35	68	62	82	88	0	0	0	0
2	39	41	28	42	44	30	47	29	33	94	109	64	0	0	0	0
3	42	28	26	48	35	46	33	42	46	103	45	106	0	0	0	0
4	37	39	35	38	41	21	48	41	52	81	66	101	0	0	0	0
5	32	34	41	37	38	28	50	40	55	31	146	68	0	0	0	0
6	35	38	32	35	46	43	36	35	87	63	74	76	0	0	0	0
7	45	33	37	34	45	24	38	44	97	48	61	94	0	0	0	0
8	41	32	36	37	33	32	44	45	62	55	78	105	0	0	0	0
9	30	41	37	47	43	33	33	36	88	107	57	48	0	0	0	0
10	31	44	36	33	44	32	43	37	64	86	97	53	0	0	0	0

Table 6: Simulations: Exploration-Exploitation. Incidence data.

	HPY-TS								Good-Turing							
Runs/Zipf	1.3	1.3	1.3	1.3	2	2	2	2	1.3	1.3	1.3	1.3	2	2	2	2
1	5	5	6	14	0	0	0	0	3	8	4	9	2	1	2	1
2	5	14	7	4	0	0	0	0	18	2	2	4	1	1	1	1
3	6	6	9	9	0	0	0	0	2	6	2	16	1	1	1	1
4	9	8	1	12	0	0	0	0	3	3	1	18	2	1	1	1
5	1	6	8	15	0	0	0	0	3	3	5	12	2	1	2	2
6	1	9	4	16	0	0	0	0	3	16	3	3	2	1	1	1
7	4	9	9	8	0	0	0	0	2	1	2	21	1	1	1	1
8	5	6	12	7	0	0	0	0	7	2	14	3	1	1	1	1
9	6	7	9	8	0	0	0	0	27	1	1	1	0	0	0	0
10	4	6	8	12	0	0	0	0	4	3	2	16	1	2	1	1
	Uniform								Oracle							
Runs/Zipf	1.3	1.3	1.3	1.3	2	2	2	2	1.3	1.3	1.3	1.3	2	2	2	2
1	7	2	0	1	7	7	2	4	8	8	7	7	0	0	0	0
2	5	2	4	2	4	3	5	5	7	7	11	5	0	0	0	0
3	4	7	3	3	2	5	6	0	7	5	12	6	0	0	0	0
4	6	4	4	1	2	3	6	4	7	3	10	10	0	0	0	0
5	4	3	4	6	3	3	3	4	8	8	9	5	0	0	0	0
6	4	2	2	6	5	6	4	1	10	6	8	6	0	0	0	0
7	1	3	2	5	9	4	5	1	10	5	10	5	0	0	0	0
8	2	5	2	0	3	8	5	5	5	11	6	8	0	0	0	0
9	4	4	3	5	4	5	2	3	5	8	10	7	0	0	0	0
10	7	4	3	1	2	4	3	6	5	9	8	8	0	0	0	0