UNIVERSITY OF TORINO

M.Sc. in Stochastics and Data Science

Final dissertation

Multi-armed bandit for species discovery: An application to Political Polls

Supervisor: Stefano Favaro

Candidate: Federico Ferrari

ACADEMIC YEAR 2016/2017

Contents

1 Introduction

Species sampling problems has been intensely studied for a long time in ecology and biology, and today they are drawing interest in other fields like machine learning, linguistics and genetics. The setting has very well-defined features and the problem has a sequential nature. First of all, imagine a set of J populations, indexed with the subscript $j = 1, 2, ..., J$, consisting of different species, as for example animals. Each animal can live in one o more areas. Formally, the supports of the populations have non-empty intersections. In addition, the frequency of each species can vary across populations. The populations could represents geographical areas as well as DNA sequences.

Our objective is to deduce the composition of the populations, together with the inference on the total number of species present. For this reason, we are given the possibility to explore them. Prior to starting the exploration, we obtain an initial sample from each population. Afterwards, the aim is developing a strategy that make us inferring the distribution of the species and learn the differences across populations. Hence, suppose that at each instant of time we can choose in which population randomly sample a unit. Then, after observing the species of the observation, we exploit the updated information set to choose the population at the following step. Everything cound be generalized to the setting in which we sample $R > 1$ units at each point in time, this is developed in [2].

The starting point of this thesis is the paper *"Multi-armed bandit for species discovery: A Bayesian nonparametric approach" (Battiston, Favaro and Teh)*. Indeed, I will firstly consider the problem of formulating a strategy that maximizes the highest number of species, or equivalently tries to sample from the population with the highest miss mass. For this purpose, Battiston, Favaro and Teh developed a Bayesian non parametric approach using a Thompson-sampling together with a Hierarchical Pitman-Yor process. I will describe their result in **Section 2.1** after a brief literature review of the main features of the PY process and its Hierarchical version. The main difference with the first version of the algorithm of [2] is the development of a Filtering procedure to estimate the hyperparamers of the HPY process. I will give a deep description of the implementation of the Filtering algorithm by Liu and West [8] in **Section 2.3** and the corresponding Matlab codes can be found in **Appendix B**. Likewise, I will present the UCB strategy in **Section 2.2** elaborated by Bubeck, Ernst and Garivier in the paper *"Optimal discovery with*

probabilistic expert advice: finite time analysis and macroscopic optimality". This latter approach exploits the Good-Turing estimator which first appeared in [7].

The nature of the problem resembles the setting of Multi-armed bandits. Indeed, the populations may represent slot machines and we need to design a winning strategy as we are gamblers. At each time, we can choose which arm playing next and we receive a reward according to an unknown Bernoulli distribution whenever we sample a new species. This is the reason why the terms "arm" and "population" will be used with the same meaning.

Afterwards, I will consider the problem of sampling species observed $k > 0$ times, with A_k the set containing those species. Thanks to the flexibility of the TS-HPY strategy elaborated in [2] for the missing mass scenario, it will be enough to find the joint posterior distribution of the populations for the set A_k Finally, I will adapt the algorithm for estimating the missing mass to this more general framework in **Section 3.2**. On the other hand, to generalize the UCB strategy, I firstly find an upper bound for the GT estimator $\tilde{M}_k$ in **Section 3.1**, following the steps of Bubeck, Ernst and Garivier in the miss mass scenario. Then, I i will use a tuning constant C that is a function of the upper bound to construct a UCB algorithm. Differently from [1], I will not carry on a regret analysis, as the UCB strategy is needed to be a term of comparison with the TS-HPY strategy.

Throughout this thesis, I will carry on several simulations using the three different scenarios described in [2]: Pure exploration, Exploration plus Exploitation and Pure exploitation. A description of these scenarios is in **Section 2.4**.

Finally, in **Section 4**, I will apply the different strategies to real data coming from electoral polls. The data are monthly surveys conducted by the Pew Research center and they can be downloaded at *http://www.pewresearch.org/data/*. I will crossclassify respondents to the questionnaire as different species according to attributes like Sex, Race and Education. In a secondary analysis I will also take into account the unlikely voters and the undecided ones. In both scenarios it is evident that the TS-HPY algorithm outperforms the other strategies.

Special acknowledgements must go to my Supervisor Stefano Favaro and to Federico Camerlenghi for having guided me during this work. I am also really thankful to my entire family, in particular my sister, my mother and my father, and to Sarah that never stopped encouraging me during my master degree, including when I chose to purse a PhD abroad. I also need to show great appreciation to Lorenzo

"Pupetto" Novello for the protracted talks of priceless value about the various sides of heterogeneity and homogeneity.

2 Missing mass estimation

In this section I describe the main algorithms for the missing mass estimation: TS-HPY, UCB, Oracle and Uniform. The setting considered is the one developed in [2]. Let $(P_1, ..., P_J)$ denote J populations of animals of distinct regions. Each population P_j is assumed to contain a large number of species of animals, but both the represented species and their frequencies are unknown a priori. Also, the J populations are allowed to share the same species of animals and each species can have different frequencies in distinct regions. In particular, here I consider the problem of sequentially sampling these populations with the goal of maximizing the number of distinct species observed.

The challenge of selecting the population that maximazes the probability of discovering a new species can be seen as a Multi-armed bandit problem. We can imagine J slot machines with unknown reward distribution, representing the populations, and at each time step we receive a gain if a new species is sampled.

In order to presend the procedures used in this work, we need to introduce Thompson sampling. TS was initially proposed by Thompson in as a randomized Bayesian algorithm to minimize regret in a clinical trial setting. The idea is to assume a prior for the unknown parameters in the distributions of each arm and, at every time step, play an arm according to its posterior probability of being the best one. Its most popular application is for Bernoulli bandits. In this setting, a Bernoulli distribution with unknown parameter is set as reward distribution for each arm, and the unknown reward means are endowed with Beta prior distributions.

Similarly, in our setting we have a stochastic formalization of the MAB problem, meaning that we are uninformed about the reward distribution. Altought, the key difference is that the distributions are changing over time as non-increasing functions of the number of draws. For this reason we have to found the joint posterior distribution for this J Bernoulli means.

2.1 Pitman-Yor process

In order to understand the strategy we are going to develop to solve the MAB problem, we need to describe the Pitman-Yor process that was introduced by Pitman and Yor in [13] as a generalization of the Dirichlet process of Ferguson. Like the Dirichlet process, the PY process is a random probability measure that assigns probability 1 to the set of discrete distributions. It is parametrized by $(\sigma, \theta, P0)$, where P_0, called the base distribution, is a distribution on the sample space, and σ and θ are two scalars satisfying $0 \leq \sigma < 1$ and $\theta > -\sigma$, respectively called the concentration and the mass parameter. The Dirichlet process corresponds to the special case $\sigma = 0$.

The PY process P admits a stick-breaking representation. Let $(Y_i^*)_{i\geq 1}$ be i.i.d. random variables distributed as P_0 and let $(p_i)_{i\geq 1}$ be weights such that $p_i = V_i \prod_{k=1}^{i-1}(1 - V_i)$ with V_i being independent with distribution $beta(1 - \sigma, \theta + i\sigma)$. Then the following random probability mesure is a Pitman-Yor process:

$$P = \sum_{i\geq 1} p_i \delta_{Y_i^*} \tag{1}$$

The posterior distribution given the observations was derived by Pitman in [12]. Given a sample $\mathbf{Y_n} = (Y_1, ..., Y_m)$, such that $Y_i|P \overset{iid}{\sim} P$ and $P \sim PY(\sigma, \theta, P_0)$, then the posterior of P given $\mathbf{Y_n}$ satistifies the following distributional equation:

$$P|\mathbf{Y_n} \sim \sum_{i=1}^{K_n} \omega_i \delta_{Y_i^*} + \omega_0 \tilde{P} \tag{2}$$

where K_n is the number of distinct values in the sample $\mathbb{Y}_n$, denoted by $(Y_1^*, ..., Y_{K_n}^*)$ and having multiplicities $(n_1, ...n_{K_n})$. $(\omega_0, ..., \omega_{K_n})$ is a random vector distributed as $Dir(\theta + K_n\sigma, n_1 - \sigma, ..., n_{K_n} - \sigma)$ and $\tilde{P} \sim PY(\sigma, \theta + K_n\sigma, P_0)$.

It is also possible to derive the so-called Chinese restaurant representation of the PY process, which describes the distribution of the next observation Y_{n+1} given the hyperparameters, the base measure P_0 and the vector of observations $\mathbb{Y}_n$:

$$Y_{n+1}|\mathbb{Y}_n, P_0, \sigma, \theta \sim \sum_{i=1}^{K_n} \frac{n_i - \sigma}{\theta + n} \delta_{Y_i^*} + \frac{\theta + K_n\sigma}{\theta + n} P_0 \tag{3}$$

The new observation takes an old value $\delta_{Y_i^*}$ with probability proportional to $n_i - \sigma$, or it is sampled from P_0 with probability proportional to $\theta + K_n\sigma$. If P_0 is

almost surely continuos, then the new observations takes always a value not yet sampled, otherwise it may take values of "old" clusters. This scheme is called "Chinese restaurant" representation because we can think of the observations as customers entering in a restaurant sequentially. As the first customer comes in, the waitress opens a new table and the individual orders dish Y_1^*. The second customer will sit at the first table with probability proportional to $1 - \sigma$ and eat dish Y_1^*, or ask the waitress to open a new table with probability proportional to $\theta + \sigma$. This process goes on until an infinite number of customers enters in the restaurant. The tables represents the clusters and their number will go to infinity as the number of customers continues to grow: the imaginary Chinese restaurant may accomodate infinite customers opening infinite tables.

Finally, the PY can be described through a hierarchical representation:

$$P|P_0, \sigma, \theta \sim PY(\sigma, \theta, P_0)$$
$$Y_i|P \overset{iid}{\sim} P \quad i = 1, 2, ..., K$$

2.1.1 HPY-TS process for missing mass estimation

We now move to the specific application of the PY process relevant for our problem, in particular we will use the Hierarchical Pitman-Yor process. Recall that one of the distinctive feautures of the Multi-armed bandit problem is that the populations are sharing atoms but also that could have unique species in them. We can think of the generating process as coming from an hyperdistribution that first generates the totality of species and then spreads them across different populations. This property can be well included into a Bayesian Nonparametric hierarchical model, as the HPY process. The HPY process is defined by the following hierarchical representation:

$$P_0|\alpha, \gamma, H \sim PY(\alpha, \gamma, H)$$
$$P_j|\sigma_j, \theta_j, P_0 \sim PY(\sigma_j, \theta_j, P_0) \quad \forall j = 1, 2, ..., J$$
$$Y_{j,i}|P_j \overset{iid}{\sim} P_j \quad \forall j = 1, 2, ..., J \quad \forall i = 1, 2, ...n_j$$

To represent the HPY process, in addition to the Chinese restaurant process for every single restaurant, we also have a metaphor for the Chinese restaurant franchise. Let n_{jtk} denote the number of observations in population j belonging to cluster t and having value Y_k^{**}, while m_{jk} is the number of clusters in population j with value Y_k^{**}. One should think of m_{jk} as the number of tables in restaurant j with value Y_k^{**}.

As in [16], the dots in the indeces denote that we are summing over that index, e.g. $m_{j\cdot} = \sum_k m_{jk}$ is the total number of clusters in population j.

Differently from the PY process, when using the hierarchical model we also have a predictive distribution for the new cluster/tables. Thanks to the exchangeability of the process, we can consider the predictive distribution for the last cluster of restaurant J:

$$Y^*_{J,m_{J+1}}|Y^*_{1,1},...,Y^*_{J,m_{J+1}},\alpha,\gamma,H \sim \sum_{k=1}^{K}\frac{m_{\cdot k}-\alpha}{\gamma+m_{\cdot\cdot}}\delta_{Y^{**}_k}+\frac{\gamma+K\alpha}{\gamma+m_{\cdot\cdot}}H \tag{4}$$

The interpretation for the clusters is similar to the one for the customers: the new cluster in population J has the same value as one already observed in the joined sample with probability proportional to $m_{\cdot\cdot} - K\alpha$ and it takes a new value, sampled from H, with probability proportional to $\gamma + K\alpha$.

Going back to the missing mass estimation, denote as $Y_{n_{j\cdot\cdot}} = (Y_{j,1}, Y_{j,2}, ..., Y_{j,n_{j\cdot\cdot}})$ the vector of observations from the j-th population, with $\mathbf{Y_n} = (\mathbf{Y}_{n_{1\cdot\cdot}}, \mathbf{Y}_{n_{2\cdot\cdot}}, ..., \mathbf{Y}_{n_{J\cdot\cdot}})$ being the joint sample, that is, the array containing the observations from all the populations. Recall that our aim is to maximize the number of different species discovered, meaning that we would like to sample species from the set: $A = \{y \in \mathbb{Y} : y \notin \mathbf{Y_n}\}$, which is the set of possible new species. Therefore, what we need is the distribution of:

$$(P_1(A), P_2(A), ..., P_J(A))|\mathbf{Y_n}, \sigma_1, ..., \sigma_J, \theta_1, ..., \theta_J, \alpha, \gamma, H$$

In the following proposition which appears in [2], Battiston-Favaro and Teh provide the density of the joint distribution of interest. To simplify the notation, I omit the conditioning on $\sigma_j, \theta_j, \alpha, \gamma, H$.

Proposition 2.1. *Let* $\mathbf{Y_n}$ *denote the joined sample from a HPY process and let* $A = \{y \in \mathbb{Y} : y \notin \mathbf{Y_n}\}$. *Then,* $(P_1(A), P_2(A), ..., P_J(A))|\mathbf{Y_n}$ *admits the following multivariate density:*

$$f_{(P_1(A),..,P_J(A))|\mathbf{Y_n}}(p_1, ..., p_J) = \int_0^1 \prod_{j=1}^{J} f_j(p_j|\beta_0, m_{j\cdot}, n_{j\cdot\cdot}) f_0(\beta_0|K, m_{\cdot\cdot})d\beta_0 \tag{5}$$

where:

$$f_j(p_j|\beta_0, m_{j\cdot}, n_{j\cdot\cdot}) = beta(p_j|(\theta_j + m_{j\cdot}\sigma_j)\beta_0, (\theta_j + m_{j\cdot}\sigma_j)(1-\beta_0) + n_{j\cdot\cdot} - \sigma_j m_{j\cdot}) \tag{6}$$

$$f_0(\beta_0|K, m_{\cdot\cdot}) = beta(\beta_0|\gamma + K\alpha, m_{\cdot\cdot} - \alpha K) \tag{7}$$

A direct application of the previous proposition may provide us with a point estimator such as the posterior mean. This is found again in [2] and it is equal to:

$$\mathbb{E}[P_j(A)|\mathbf{Y_n}] = \Big(\frac{\theta_j + m_{j.}\sigma_j}{\theta_j + n_{j..}}\Big)\Big(\frac{\gamma + K\alpha}{\gamma + m_{..}}\Big) \tag{8}$$

Nevertheless, the HPY Thompson-sampling strategy prescribes to sample a draw from (2) and (3) and the select the population with the highest realized value p_j. This is done in order to better balance the exploration step. The strategy for HPY-TS is summarized in the following algorithm:

Algorithm 1 HPY-TS Algorithm for the missing mass

for j in $1 : J$ **do**
  Sample $\frac{n_{init}}{J}$ units from each population
end for
Calculate table counts and estimate the HPY hyperparameters

for i in 1:additional sample **do**
  Draw $\beta_0 \sim beta(\gamma + K\alpha, m_{..} - \alpha K)$
  for j in 1:J **do**
    Draw $p_j \sim beta((\theta_j + \sigma_j m_{j.})\beta_0, (\theta_j + \sigma_j m_{j.})(1 - \beta_0) + n_{j..} - \sigma_j m_{j.})$
  end for
  Compute $j^* = argmax\{p_j : j = 1, 2, ..., J\}$
  Sample next observation from population j^*
  Update table counts and estimates of the HPY hyperparameters
end for

2.1.2 HDP-TS

For comparison purposes, I decided to try also the Hierarchical Dirichlet (HDP) process to address the multi-armed bandit problem. What follows is simply a proposition that mimics the one in the supplementary material of [2] which can be found online. As we will see, the HDP works slightly worse that the HPY, but in any case better than the UCB and Uniform strategies. Of course, we will use the following

result to create a strategy to solve the Multi-armed bandit problem. The algorithm is presented in the appendix.

Proposition 2.2. *Let* $\mathbf{Y_n}$ *denote the joined sample from a HDP and let* $A = \{y \in \mathbb{Y} : y \notin \mathbf{Y_n}\}$. *Then,* $(P_1(A), P_2(A), ..., P_J(A))|\mathbf{Y_n}$ *admits the following multivariate density:*

$$f_{(P_1(A),..,P_J(A))|\mathbf{Y_n}}(p_1, ..., p_J) = \int_0^1 \prod_{j=1}^{J} f_j(p_j|\beta_0, m_{j.}, n_{j..}) f_0(\beta_0|K, m_{..}) d\beta_0 \quad (9)$$

where:

$$f_j(p_j|\beta_0, m_{j.}, n_{j..}) = beta(p_j|\theta_j\beta_0, \theta_j(1-\beta_0) + n_{j..}) \quad (10)$$

$$f_0(\beta_0|K, m_{..}) = beta(\beta_0|\gamma, m_{..}) \quad (11)$$

Proof

Given a sample $\mathbf{Y_n} = (Y_1, ..., Y_n)$ such that $Y_i|P \overset{iid}{\sim} P$ for $i = 1, 2, ...n$ and $P \sim DP(\theta, H)$, the posterior of P given $\mathbf{Y_n}$ satisfies the following distributional equation:

$$P|\mathbf{Y_n} \overset{d}{=} \sum_{k=1}^{K_n} \omega_i \delta_{Y_i^*} + \omega_0 P \quad (12)$$

where K_n is the number of distinct values in the sample. Let $(n_1, n_2, ...n_{K_n})$ be their multiplicities, We have that:

$$(\omega_0, \omega_1, ..., \omega_{K_n}) \sim Dir(\theta, n_1, ..., n_{K_n}) \quad (13)$$

Let's first apply equation (1) to the hyperprior P_0 with the conditioning on the overall sample. Remember that $Y_1^{**}, ..., Y_K^{**}$ are the distinct value franchise-wide and $m_{.k}$ the number of tables serving dish Y_k^{**}.

$$P_0|\mathbf{Y_n} \overset{d}{=} \sum_{k=1}^{K} \beta_i \delta_{Y_i^{**}} + \beta_0 P$$

$$(\beta_0, \beta_1, ...\beta_K) \sim Dir(\gamma, m_{.1}, ..., m_{.K})$$

Now, we need to find a distributional equation for P_j conditioning on data, P_0 and on $\beta = (\beta_1, \beta_2, ...\beta_K)$. Similarly we apply formula (1) and we take advantage of the additive property of the Dirichlet distribution.

$$P_j|P_0, \mathbf{Y_n}, \beta \overset{d}{=} \sum_{k=1}^{K} \pi_{j,k} \delta_{Y_i^{**}} + \pi_{j,0} P_j$$

$$\pi = (\pi_{j,0}, \pi_{j,1}, ..., \pi_{j,K})|\beta, \mathbf{Y_n} \sim Dir(\theta_j\beta_0, \theta_j\beta_1 + n_{j.1}, ..., \theta_j\beta_K + n_{j.K}) \quad (14)$$

Now, we want to calculate $P_j(A)|P_0, \mathbf{Y_n}, \beta$. First of all $\delta_{Y_k^{**}=0}\ \forall k = 1, 2, ..., K$ since $\{Y_1^{**}, Y_2^{**}, ..., Y_K^{**}\} = A^c$. Also, $P_j(A) = 1$, so we have that:

$$\begin{aligned} P_j(A)|P_0, \mathbf{Y_n}, \beta &\stackrel{d}{=} \pi_{j,0}|\mathbf{Y_n}, \beta \\ &\stackrel{d}{=} beta(\theta_j\beta_0, \theta_j(1-\beta_0) + n_{j..}) \end{aligned}$$

Where for the last step we used again the aggregation property of the Dirichlet distribution. To find the joint distribution, one has just to integrate out β_0 and exploit the conditional independence of the P_j's given the measure P_0. In **Section 2.4** I compare the HPY-TS and HDP-TS together with UCB, Oracle and Uniform that I describe in **Section 2.2**.

2.2 UCB

Let us concentrate in developing a strategy that maximizes the total number of different species discovered. A possible solution could be built upon the Good-Turing estimator. This estimator first appeared in [1], and it is useful to evaluate of the probability of observing a unit that has frequency k in the sample. I.e., $\mathbb{P}(X_{n+1}\ \textit{is a species with frequency}\ k|X_1, ..., X_n)$, where X_n is the random variable describing the species of the n^{th} sampled unit. The Good-Turing estimator for a given frequency k is:

$$\tilde{M}_k = \frac{M_{k+1}}{n}$$

Where M_{k+1} is the number of species with frequency $k+1$ in the overall sample of size n. With respect to units not yet observed, we refer to estimation of the missing mass. The corresponding Good-Turing estimator is of course $\tilde{M}_0 = \frac{M_1}{n}$.

Thereafter, I will describe the Upper Confidence Bound (UCB) strategy that has been developed by Bubeck, Ernst and Garivier in [1]. This strategy exploits the Good-Turing estimator to produce a point estimate of the missing mass, then adds a deterministic upper bound. The population that we will choose to explore is the one with the highest point estimate plus the upper bound. *Proposition 1* provides a computation of the above mentioned upper bound.

Proposition 2.3. *Let* $R_0 = \mathbb{P}(X_{n+1} \text{ is a species not yet observed})$ *and* $\tilde{M}_0$ *the relative Good-Turing estimator. Then with probability at least* $1 - \delta$

$$\tilde{M}_0 - \frac{1}{n} - (1+\sqrt{2})\sqrt{\frac{log(4/\delta)}{n}} \leq R_0 \leq \tilde{M}_0 + (1+\sqrt{2})\sqrt{\frac{log(4/\delta)}{n}} \qquad (15)$$

Section 3.1 provides a proof of the statement, while I now focus on the properties of the upper bound. As we can see, the upper bound is a function of a tuning constant C and of the sample size n. The costant that is choosen in [1] to implement the UCB strategy is actually slightly different, namely $C = (1+\sqrt{2})\sqrt{3}$, which is the one that minimizes the expected regret. The regret is defined as the difference between the strategy's payoff with an 'Oracle' strategy. The 'Oracle' strategy chooses the arm with the highest expected payoff, since the uncertainness about the reward distributions is removed. The constant C is convenient to better balance the so-called exploration step. Essentially, as the upper bound is inverserly proportional to the number of times that the arm has been played, the UCB strategy may choose an arm in which the estimate of the missing mass is low with respect to other arms, but that has been less explored. The UCB algorithm for the missing mass is written explictly in the appendix.

We compare the results obtained by UCB and HPY-TS with two benchmark strategies: Uniform and Oracle. The former simply selects at each time step a certain population with probability equal to $\frac{1}{J}$. On the other hand, the Oracle strategy knows exactly the distributions that have generated the data. Hence, at each time, it chooses the population with the highest missing mass. The algorithm for the Oracle strategy can be found in the appendix.

2.3 Bootstrap-Filtering

The Multi–armed bandit problem is a sequential allocation problem, in which one has to find a strategy to sequentially sample new individuals from J different populations. As a consequence the hyperparameters referring to the HYP process must be updated whenever a new observation is sampled. A possible way to handle this problem has been first suggested in [1], here the authors employ an MCMC algorithm to estimate all the parameters. This approach is quite slow, since every time a new individual is sampled one has to run an MCMC procedure. In this section, we suggest a totally different approach to update the hyperparameters in order to

speed up the whole sequential algorithm. Such an approach has been developed in an ongoing project with F. Camerlenghi and S. Favaro, and is based on the filtering algorithm contained in [8].
The key ingredient of our approach consists in approximating the posterior distribution of the hyperparameters through a mixture of Gaussian kernels. Such an approximation allows us to use the posterior at time t to obtain an approximation of the posterior at time $t+1$, which is used to update the hyperparameters.
More precisely, here we focus on a general model which may be specified by the following probability evolving in time

$$p(y_t|\theta) \tag{16}$$

where y_t is the unit sampled at time t and θ the vector containing the hyperparameters. Besides we define the information available at time t as $D_t = \{y_t, D_{t-1}\}$.
At each time, once we observe y_{t+1}, we are interested in producing a sample from the posterior $p(\theta|D_{t+1})$. Let's apply Bayes Theorem to this posterior density:

$$p(\theta|D_{t+1}) \propto p(y_{t+1}|\theta, D_t)p(\theta|D_t) \tag{17}$$

where $p(\theta|D_t)$ the prior density for θ at time t. Actually, the filtering algorithm implemented by [8] can address the sequential updating for a wider class of models. What it is missing here with respect to the general case is the state vector x_t. To have a glimpse on this, note that the filter could work for models like

$$p(y_t|x_t, \theta)$$
$$p(x_t|x_{t-1}, \theta)$$

Where the state equation $p(x_t|x_{t-1}, \theta)$ describes the evolution of the states through a Markovian process.
Let us now focus our attention on the appoximation of the posterior density $p(\theta|D_{t+1})$. At time t we have an importance sample of $\{\theta_t^{(i)} : i = 1, 2, ..., N\}$ with a set of weights $\{\omega_t^{(1)}, ..., \omega_t^{(N)}\}$. Remember that the parameter θ does not depend on time: the index refers to the importance sample we have a time t. The smooth kernel density of [20] and [21] is given by:

$$p(\theta|D_t) \approx \sum_{i=1}^{N} \omega_t^{(i)} N(\theta|m_t^{(i)}, h^2\mathbf{V}_t) \tag{18}$$

where $\mathbf{V}_t$ is the Monte Carlo posterior variance, $\mathbf{m}_t^{(i)}$ the location of the multivariate normal density that has to be defined, and h the smoothing parameter.
First of all, it seems natural to set $m_t^{(i)} = \theta_t^{(i)}$. Nevertheless, the resulting variance of the density we are trying to estimate boils down to $(1 + h^2)\mathbf{V}_t$, which is larger than $\mathbf{V}_t$. Hence, we will make an over-dispersed approximation of θ, which represents a "loss of information" that we will carry on over time.
To solve this problem, West in [20] and [21] suggests the idea of shrinkage of kernel locations:

$$m_t^{(i)} = a\theta_t^{(i)} + (1 - a)\bar{\theta}_t \tag{19}$$

With $a = \sqrt{1 - h^2}$ and $\bar{\theta}_t$ the mean of the Monte carlo sample, we are able to preserve the variance $\mathbf{V}_t$ over time.
An improvement of the algorithm described above relies on adding independent and zero-mean normal increments to the vector of parameters at each time step. This extension, that we do not implement aims at generating at each time new parameter values, in order to avoid getting stuck with the same set of values.
Now, let's combine equations (17) and (18) and move the likelihood inside the summation:

$$p(\theta|D_{t+1}) \propto \sum_{i=1}^{N} \omega_t^{(i)} p(y_{t+1}|\theta, D_t) N(\theta|m_t^{(i)}, h^2\mathbf{V}_t) \tag{20}$$

Then, the algorithm in [8] prescribes to compute $m_t^{(i)}$ and substitute it in the likelihood, so that equation (21) now reads:

$$p(\theta|D_{t+1}) \propto \sum_{i=1}^{N} \omega_t^{(i)} p(y_{t+1}|m_t^{(i)}, D_t) N(\theta|m_t^{(i)}, h^2\mathbf{V}_t) \tag{21}$$

At this point, we have to evaluate the weights $g_{t+1}^{(i)} \propto \omega_t^{(i)} p(y_{t+1}|m_t^{(i)}, D_t)$. A large value of $g_{t+1}^{(i)}$ indicates that $m_t^{(i)}$ is likely to be more consistent with y_{t+1} with respect to a lower value.
New indicators k are now sampled with probabilities proportional to $g_{t+1}^{(i)}$. Moreover, a new parameter vector $\theta_{t+1}^{(k)}$ is sampled from the k^{th} normal component of the kernel density:

$$\theta_{t+1}^{(k)} \sim N(\cdot|m_t^{(k)}, h^2\mathbf{V}_t)$$

Summing up, we can describe the general filtering procedure depicted above as follows in the next page.

Finally, it remains to clarify the meaning of the vector of parameter θ and of the conditional distribution $p(y_{t+1}|\theta^{(k)}_{t+1}, D_t)$ in our setting. The vector θ contains the hyperparameters referring to the random probability measures $P, P_1, P_2, ..., P_J$, that is to say:

$$\theta = (\alpha, \gamma, \sigma_j, \theta_j : j = 1, 2, ...J)$$

Recall that in Multi-armed bandit problems one first assumes to be provided with an initial sample for each population. Hence one has to run an MCMC procedure based on the CRF metaphor to estimate both the table counts and the parameters θ. As for the latter, we consider the N_{iter} outputs of the MCMC and denote them by $\{\theta^{(k)}_0 : k = 1, 2, ...N_{iter}\}$. This may be regarded as an importance sample of θ with equal weights $\omega^{(k)}_0 = \frac{1}{N_iter}$ at time $t = 0$.

Moving forward, y_{t+1} represents the information added by the new customer. Referring to the chinese franchise representation of the HPY, the random variable y_{t+1} describes the dish eaten by the customer entering in a certain restaurant at time $t+1$. In our setting, the conditional distribution $p(y_{t+1}|\theta^{(k)}_{t+1}, D_t)$, may be recovered from the expression of the EPPF, which amounts to be

$$EPPF(m_{t1}, ..., m_{tK}|\alpha^{(k)}_{t+1}, \gamma^{(k)}_{t+1}) \prod_{j=1}^{J} EPPF(n_{j\cdot 1}, ..., n_{j\cdot k_j}|\theta^{(k)}_{j,t+1}, \sigma^{(k)}_{j,t+1}) \quad (22)$$

thanks to the independence across different restaurants and $EPPF$ denotes the Exchangeable partition probability function. Recall that $n_{j\cdot k}$ is the sample sizes of the "customers" eating dish $k = 1, 2, ..., k$ in restaurant j, according to the chinese restaurant representation. To clarify the notation, observe that $\theta^{(k)}_{j,t+1}$ is a the k^{th} concentration parameter of the importance sample at time $t + 1$ for restaurant j. Besides recall that the Exchangeable Partition function (EPPF) in the case of the Pitman-Yor process with cluster frequencies $n_1, n_2, ...n_k$ is:

$$\mathbb{P}(n_1, n_2, ..., n_k, K_n = k|\theta, \sigma) = \frac{\prod_{i=1}^{k-1}(\theta + \sigma j)}{\theta_{(n)}} \prod_{i=1}^{k}(1 - \alpha)_{(n_i - 1)}$$

where $\theta_{(n)} := \Gamma(n + \theta)/\Gamma(\theta)$ is the Pochhammer symbol. Since the generalized factorial coefficients and the gamma function assume high numerical values, it is better to deal with the logarithm of (22) in order to avoid numerical issues in the real implementation.

Algorithm 2 Filtering algorithm for the updating of hyperparameters

A) Evaluate $m_t^{(i)}$ for each $i = 1, 2, ..., N_{iter}$ that are the prior point estimates of θ where:

$$m_t^{(j)} = a\theta_t^{(j)} + (1-a)\bar{\theta}_t$$

B)

for i in 1:N_{iter} **do**

1) Sample an auxiliary integer variable k from the set $\{1, 2, ..., N_{iter}\}$ with probability proportional to:

$$g_{t+1}^{(i)} \propto p(y_{t+1}|m_t^{(i)}, D_t)$$

2) Sample a new parameter vector $\theta_{t+1}^{(k)}$ from the k^{th} normal component of the kernel density, namely:

$$\theta_{t+1}^{(k)} \sim N(\cdot|m_t^{(k)}, h^2\mathbf{V}_t)$$

3) Evalutate the corresponding weight

$$\omega_{t+1}^{(k)} \propto \frac{p(y_{t+1}|\theta_{t+1}^{(k)}, D_t)}{p(y_{t+1}|m_{t+1}^{(k)}, D_t)}$$

end for

C) Resample according to the importance weights $\omega_{t+1}^{(k)}$ to obtain a set of parameters with equal weights, or in other words a direct Monte Carlo approximation of the posterior.

2.4 Simulations

In this section, I describe the setting of the simulations in [2], The number of populations J is fixed and it is equal to 8. Also the true distribution of each arm is supported on a subset of size 2500, allowing for a partial sharing of the supports. The populations are assumed to follow Zipf laws. The mass assigned to the k^{th} most common species in population j, is:

$$p_j(k; s_j) = \frac{1/k^{s_j}}{\sum_{n=1}^{2500}(1/n^{s_j})}$$

Likewise, it is beneficial to recall that $p_j(k; s_j)$ has a Zipf distribution if the the number of elements with a given frequency f has a Power law distribution with

$$p(f) = \alpha f^{-1-\frac{1}{s_j}} \tag{23}$$

This property will be helpful later on for visualization purposes. The parameter s_j controls the spread of the probability mass along the atoms. As s_j decreases, the mass will be less concentrated on a low number of atoms. Hence, a good strategy should try to discover as soon as possible which arms have a low s_j and afterwards explore them. We will refer the arms with lowest s_j as "winning" ones.

At the beginning of the algorithm, we sample 30 units in each population, and then at each time step the different strategies described in the previous sections select independently the population to explore. Rembember that the aim is finding the highest number of different species. We then differentiate between three different scenarios:

1. Pure Exploitation - Zipf parameters $[1.3, 1.3, 2, 2, 2, 2, 2, 2]$
   In this setting a winning strategy should stop exploring as soon as it finds the two winning arms that have Zipf parameter equal to 1.3.

2. Pure Exploration - Zipf parameters $[1.3, 1.3, 1.3, 1.3, 1.3, 1.3, 2, 2]$
   In this setting a winning strategy should keep on exploring the six winning arms without getting stuck in any of them.

3. Exploration plus Exploitation - Zipf parameters $[1.3, 1.3, 1.3, 1.3, 2, 2, 2, 2]$ In this setting a winning strategy should balance the exploration and exploitation step, finally finding the four winning arms.

Finally, in the following plot I show the performances of the strategies described so far, i.e. HPY-TS,HDP-TS, Oracle, Uniform and UCB, in the *Exploration plus Exploitation* setting.

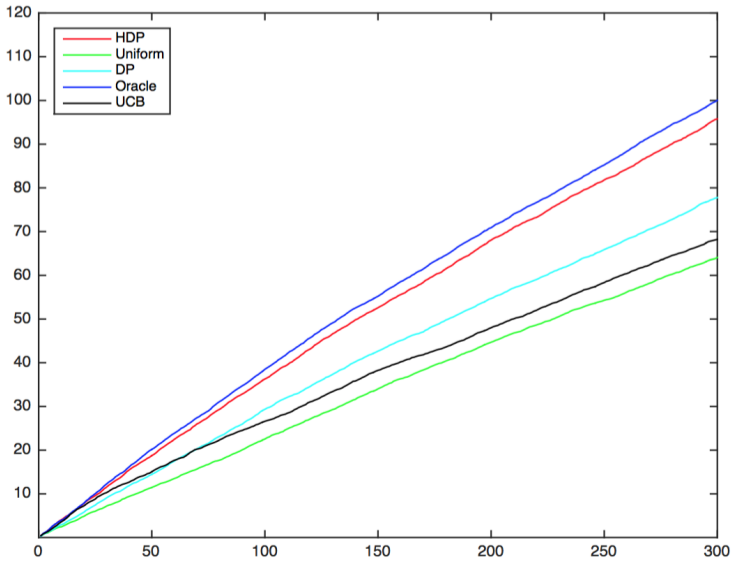


Figure 1: Performance of HPY-TS,HDP-TS, Oracle, Uniform and UCB strategies in the *Exploration plus Exploitation* setting

3 Generalization to frequencies $k > 0$

The objective of this section is to generalise the result obtained so far to species with frequency $k > 0$ in the sample. Let us again consider the previous exploration setting As before, we select at the beginning a given number of units from each population. We establish the species together with other characteristics of the discovered individuals. For example in the application we will present in **Section 4** we learn the political orientation of the respondents. Now, a possible objective could be assessing weather or not there is homogeneity within groups having the same species with respect to other variables of interest. For instance, in our application, if there is the tendency in asian-american male with a certain income to vote for the republican party or for the democrats. Hence, our next target will be resampling units with a low frequency in the initial sample.

For this purpose, we have to generalize the UCB strategy for the missing mass to every frequency $k > 0$. In the UCB setting, we follow the same steps of [1] in order to develop an upper bound between $\mathbb{P}(X_{n+1}$ is one of the individuals with frequency k in the sample) and the Good-Turing estimator $\tilde{M}_k := (k+1)\frac{M_{k+1}}{n}$ where M_k is the number of species with frequency k in the overall sample. This is done in the following **Section 3.1** through the derivation of three inequalities. Similarly, we have to modify the target set from which we want to sample units, we will call it A_k. In the online notes of [2], in *Proposition 3* the authors find an expression for the joint posterior distribution $(P_1(A_k), P_2(A_k), ..., P_J(A_k))|\mathbf{Y_n}$, and then exploits the result for writing a Thompson-HPY algorithm.

3.1 Generalized UCB

Let us now clarify several definitions that will be useful later on. First of all, $(X_1, X_2, ..., X_n)$ is the random sample of a single population. As before, K_n is the number of different species in the sample, whereas M_k is the number of different species with frequency k in the sample. Of course what we have is:

$$\sum_{k=1}^{n} M_k = K_n$$
$$\sum_{k=1}^{k} kM_k = n$$

As already pointed out, we are interested in finding an upper bound between the Good-Turing estimator and $\mathbb{P}(X_{n+1}$ is one of the type with frequency k in the sample). Hence, we define the set A_k of elements with frequency k as:

$$A_k := \{\theta \in \{\theta_1^\star, \theta_2^\star, ..., \theta_{K_n}^\star\} : \#\{i : \theta_i = \theta\} = k\} \tag{24}$$

Where $\theta_1^\star, \theta_2^\star, ..., \theta_{K_n}^\star$ are the unique species in the sample. The main result of this section is the following Theorem in which, similarly to what Bubeck, Ernst and Garivier did in [1] for the missing mass, we find an upper bound for the probability of sampling a species with a certain frequency. The proof of this theorem is given in three steps throughout this section.

Theorem 3.1. *Let* $k \in \{0, 1, 2, ..., \lfloor \frac{n}{2} \rfloor\}$, $\tilde{M}_k$ *the Good-Turing estimator,* $M_{n,k} = \mathbb{P}(X_{n+1} \in A_k)$ *with probability at least* $1 - \delta$*:*

$$M_{n,k} \geq \tilde{M}_k - \sqrt{\frac{1}{nb_k} log(\frac{8}{\delta})} - (k+1)\Big[\frac{1}{n} + \sqrt{\frac{2}{n} log(\frac{4}{\delta})}\Big] \tag{25}$$

$$M_{n,k} \leq \tilde{M}_k + \sqrt{\frac{1}{nb_k} log(\frac{8}{\delta})} + (k+1)\Big[\frac{k}{n^2 + nk} + \sqrt{\frac{2}{n} log(\frac{4}{\delta})}\Big] \tag{26}$$

where $b_k = \frac{1}{8(k+1)e^{k+1}}$ *and* A_k *is defined as before in* (16)

Let us just work a little bit with $M_{n,k}$ and the Good-Turing estimator $\tilde{M}_k$. We add and subtract terms and use basic inequalities:

$$\begin{aligned} M_{n,k} - \tilde{M}_k &= M_{n,k} - \tilde{M}_k \pm \mathbb{E}[M_{n,k}] \pm \mathbb{E}[\tilde{M}_k] = \\ &= (\mathbb{E}[M_{n,k}] - \mathbb{E}[\tilde{M}_k]) + (\tilde{M}_k - \mathbb{E}[\tilde{M}_k]) + (M_{n,k} - \mathbb{E}[M_{n,k}]) \leq \\ &\leq \underbrace{|\mathbb{E}[M_{n,k}] - \mathbb{E}[\tilde{M}_k]|}_{\text{Theorem 3.2}} + \underbrace{|\tilde{M}_k - \mathbb{E}[\tilde{M}_k]|}_{\text{BD Inequality}} + \underbrace{|\mathbb{E}[M_{n,k}] - M_{n,k}|}_{\text{Thm 5 Ohanessian-Dahleh}} \end{aligned}$$

Therefore, we will need three different inequalities. Firstly, we find a bound between $\mathbb{E}(\mathbb{P}(X_{n+1} \in A_k))$ and $\mathbb{E}\tilde{M}_k$ in the Binomial setting with *Theorem 3.2*. Then, we use the Bound and Difference inequality of McDiarmid (1989) to construct an inequality between $\tilde{M}_k$ and its expectation. Then, we use *Theorem 5* in [11] that helps us compare $\mathbb{P}(X_{n+1} \in A_k)$ and its expected value. Finally, we put everything together to get the desired inequality.

3.1.1 First Inequality in the Binomial setting

We now provide the first inequality needed to prove *Theorem 3.1*, the meaning of "Binomial Setting" will be clear as we proceed. We want to estimate $\mathbb{P}(X_{n+1} \in A_k)$, which is stochastic, since it depends on the randomness of sampling. To make this point clear, let us think at the population as almost surely discrete, i.e.:

$$\sum_{i \geq 1} p_i \delta_{Y_i}$$

Where $(Y_i)_{i\geq 1}$ are the locations of the atoms. To calculate $\mathbb{P}(X_{n+1} \in A_k)$ it will be enough to sum the p_i such that in the sample there are k units of the i^{th} species, formally:

$$\mathbb{P}(X_{n+1} \in A_k) = \sum_{i\geq 1} p_i \mathbb{1}(N_i = k)$$

where N_i is a random variable describing the number of times we sampled the i-th species. Clearly, as $\mathbb{P}(X_{n+1} \in A_k)$ is a function of the random variables N_i's, it is itself a random variable. As usual, we may estimate a random object computing its mean:

$$\begin{aligned}\mathbb{E}(\mathbb{P}(X_{n+1} \in A_k)) = \mathbb{E}(\sum_{i\geq 1} p_i \mathbb{1}(N_i = k)) = \sum_{i\geq 1} p_i \mathbb{P}(N_i = k) = \\ = \sum_{i\geq 1} p_i \binom{n}{k} p_i^k (1-p_i)^{n-k} = \\ = \sum_{i\geq 1} \binom{n}{k} p_i^{k+1} (1-p_i)^{n-k}\end{aligned}$$

Where we exploited the fact that N_i has a Binomial distribution with probability of success p_i, this is why we call it binomial setting. Recall that we try to estimate $P(X_{n+1} \in A_k)$ with the Good-Turing estimator $\tilde{M}_k$. its mean is equal to:

$$\begin{aligned}\mathbb{E}\tilde{M}_k = \frac{k+1}{n}\mathbb{E}M_{k+1} = \frac{k+1}{n}\mathbb{E}\sum_{i\geq 1} \mathbb{1}(N_i = k+1) = \\ = \frac{k+1}{n}\sum_{i\geq 1} \binom{n}{k+1} p_i^{k+1} (1-p_i)^{n-(k+1)}\end{aligned}$$

Finally, we have all the ingredients to compare $\mathbb{E}\tilde{M}_k$ and $\mathbb{E}(\mathbb{P}(X_{n+1} \in A_k)$. This is done in the following *Theorem 3.2* that provides us with the first inequality we need for proving *Theorem 3.1*.

Theorem 3.2. *In the binomial setting, assuming that an almost surely discrete distribution generates the data, we have that:*

$$\mathbb{E}[\mathbb{P}(X_{n+1} \in A_k) - \tilde{M}_k] \in \Big[-\frac{k+1}{n}, \frac{k^2+k}{n^2-nk} \Big] \tag{27}$$

where $\tilde{M}_k$ is the Good-Turing estimator for frequency k

Proof

We subtract the expressions we have calculated before for $\mathbb{E}\tilde{M}_k$ and $\mathbb{E}(\mathbb{P}(X_{n+1} \in A_k)$:

$$\mathbb{E}(P(X_{n+1} \in A_k) - \tilde{M}_k) = \sum_{i\geq 1} \binom{n}{k+1} p_i^{k+1}(1-p_i)^{n-(k+1)} \Big[\frac{\binom{n}{k}}{\binom{n}{k+1}}(1-p_i) - \frac{k+1}{n}\Big] =$$

$$= \sum_{i\geq 1} \binom{n}{k+1} p_i^{k+1}(1-p_i)^{n-(k+1)} \Big[\frac{k+1}{n-r}(1-p_i) - \frac{k+1}{n}\Big]$$

Exploiting the simplest properties of a probability measure:

$$\sum_{i\geq 1} \binom{n}{k+1} p_i^{k+1}(1-p_i)^{n-(k+1)}(1-p_i) \in [0,1]$$

$$\Big[\frac{k+1}{n-r}(1-p_i) - \frac{k+1}{n}\Big] \in \Big[-\frac{k+1}{n}, \frac{k+1}{n-r} - \frac{k+1}{n}\Big] = \Big[-\frac{k+1}{n}, \frac{r^2+r}{n^2-nr}\Big]$$

Consequently, we have found the desired inequality which also proves the Theorem:

$$\mathbb{E}(\mathbb{P}(X_{n+1} \in A_k) - \tilde{M}_k) \in \Big[-\frac{k+1}{n}, \frac{k^2+k}{n^2-nk}\Big]$$

Remark 3.3

It is interesting to notice that:

$$\sum_{i\geq 1} \binom{n}{k+1} p_i^{k+1}(1-p_i)^{n-(k+1)}(1-p_i) = 1 - \sum_{i\geq 1} p_i \binom{n}{k+1} p_i^{k+1}(1-p_i)^{n-(k+1)} =$$

$$= 1 - \mathbb{E}(\mathbb{P}(X_{n+1} \in A_{k+1})) \in [0,1]$$

Which is the reason why the inequality in **Theorem 3.1** is tight, since we expect $\mathbb{E}(\mathbb{P}(X_{n+1} \in A_{k+1}))$ to be small.

Corollary 2.4

A direct application of *Theorem 3.1* for the GT estimator $\tilde{M}_0$ of the missing mass can be used as a proof of the first part of *Proposition 1* in [1].

$$\mathbb{E}(\mathbb{P}(X_{n+1} \in A_0) - \tilde{M}_0) \in [-\frac{1}{n}, 0]$$

$$|\mathbb{E}[\tilde{M}_0] - \mathbb{E}[\mathbb{P}(X_{n+1} \text{ is a new value})]| \leq \frac{1}{n}$$

In some cases, it may ne convenient to have this expressions in the Poisson setting. Indeed, we use the Poisson distribution as limit of the binomial when $n \to \infty$ and $p_i \to 0$, while the expectation np_i converges to a constant $\neq 0$. The random variables N_i are now distributed as $Po(np_i)$. i.e.:

$$\mathbb{P}(N_i = k) = e^{-np_i} \frac{(np_i)^k}{k!}$$

At this point, we can easly check that the previous inequality between $\mathbb{E}(\mathbb{P}(X_{n+1} \in A_k))$ and $\mathbb{E}\tilde{M}_k$ is actually an equality:

$$\mathbb{E}(\mathbb{P}(X_{n+1} \in A_k)) = \sum_{i \geq 1} p_i e^{-np_i} \frac{(np_i)^k}{k!}$$

$$\mathbb{E}\tilde{M}_k = \frac{k+1}{n} \sum_{i \geq 1} e^{-np_i} \frac{(np_i)^{k+1}}{(k+1)!} = \mathbb{E}(\mathbb{P}(X_{n+1} \in A_k)$$

3.1.2 Second Inequality - McDiarmid inequality (1989)

In this section, we use the following Bound and Difference inequality (BDI) of [10] to find a bound between $\tilde{M}_k$ and $\mathbb{E}\tilde{M}_k$. The BDI is very powerful tool when dealing with Good-Turing estimators. In particulr, we will now give the statement of the inequality as appears in [10] and afterwards we apply it to $\tilde{M}_k$ in **Proposition 3.6**.

Theorem 3.5. *Let $X_1, X_2, ..., X_n$ be independent random variables, taking values in A_k for each k. Suppose that the measurable function $f : A_1 x A_2 x ....x A_n \to \mathbb{R}$ satisfies:*

$$|f(x_1, x_2, .., x_k, ..., x_n) - f(x_1, x_2, .., x'_k, ..., x_n)| \leq c_k \tag{28}$$

Let Y be the random variable $f(X_1, X_2, .., X_k, ..., X_n)$, then for any $\epsilon > 0$:

$$\mathbb{P}(|Y - \mathbb{E}(Y)| \geq \epsilon) \leq 2e^{-2\frac{\epsilon^2}{\sum_{k=1}^n c_k^2}} \tag{29}$$

Proposition 2.6

Let $\tilde{M}_k$ be the Good-Turing estimator. With probability at least $1-\delta$:

$$|\tilde{M}_k - \mathbb{E}[\tilde{M}_k]| \leq \sqrt{2log(\frac{2}{\delta})\frac{(r+1)^2}{n}} \tag{30}$$

Proof

First of all, the random variable $\tilde{M}_k$ is a function of the independent observations $X_1, X_2, ..., X_n$. Moreover, recall that our sampling distribution is almost surely discrete:

$$\tilde{M}_k = f(X_1, X_2, ..., X_n) = \frac{k+1}{n}\sum_{i\geq 1}\mathbb{1}(N_i = k+1)$$

When you modify just one observation you have a limited impact on the value of $\tilde{M}_k$, formally $\forall l \in \{1, 2, ...n\}$:

$$|f(X_1, X_2, ..., X_n) - f(X_1, X_2, .X_{l-1}, X'_l, X_{l+1}..., X_n)| \leq 2\frac{k+1}{n}$$

The worst case scenario happens when $X_l = a$ with $N_a = k+2$ and $N_b = k$ and you modify one observation putting $X'_l = b$. It is easy to see that:

$$|\tilde{M}_k - \tilde{M}'_k| = 2\frac{k+1}{n}$$

Hence the constant c_k in this case is equal to $2\frac{r+1}{n}$. A direct application of *Theorem 4.1* from McDiarmid(1989) reads:

$$\mathbb{P}(|\tilde{M}_k - \mathbb{E}(\tilde{M}_k)| \geq \epsilon) \leq 2e^{-\frac{\epsilon^2}{\sum_{k=1}^{n}(2\frac{k+1}{n})^2}} = 2e^{-\frac{\epsilon^2 n}{2(k+1)^2}} \tag{31}$$

Finally, if we set $\delta = 2e^{-\frac{\epsilon^2 n}{2(k+1)^2}}$, we get the second inequality that holds with probability at least $1-\delta$:

$$|\tilde{M}_k - \mathbb{E}[\tilde{M}_k]| \leq \sqrt{2log(\frac{2}{\delta})\frac{(k+1)^2}{n}}$$

3.1.3 Third Inequality - Theorem 5 Ohannessian, Dahleh (2012)

What remains to do is finding a relation between $P(X_{n+1} \in A_k)$ and its expectation. To do so, we exploit a result by Ohannessian and Dahleh [11] that can be used for $k < \frac{n}{2}$. We also use the notation provided by the authors, i.e. $M_{n,k} = P(X_{n+1} \in A_k)$. The following theorem provides the third and last inequality we need to prove *Theorem 3.1*

Theorem 3.3. *Consider an arbitrary* $\mathbb{P}$. *Then, for every* $k = 0, 1, 2, ...$ *such that* $n > 2k$, *and for all* $0 < \epsilon < 1$, *we have:*

$$\mathbb{P}(|M_{n,k} - \mathbb{E}[M_{n,k}]| > \epsilon) \leq 4e^{-b_k \epsilon^2 n} \tag{32}$$

Where $b_k = \frac{1}{8(k+1)e^{k+1}}$

As done in the previous section, if we set $\delta = 4e^{-b_k \epsilon^2 n}$, we find that $\epsilon = \sqrt{\frac{1}{nb_k} log(\frac{4}{\delta})}$. Formally:

$$\mathbb{P}\Big(|M_{n,k} - \mathbb{E}(M_{n,k})| > \sqrt{\frac{1}{nb_k} log(\frac{4}{\delta})}\Big) \leq \delta$$

3.1.4 Upper Confidence Bound

As the theorem can be easily proven through the three inequalities we found, what remains to do is finding the suitable upper bound for the multi-armed bandit problem. We keep using the notation of [11], i.e. $M_{n,k} = P(X_{n+1} \in A_k)$. Let us just show the upper bound of the theorem.

$$\begin{aligned}
M_{n,k} - \tilde{M}_k &= M_{n,k} - \tilde{M}_k \pm \mathbb{E}[M_{n,k}] \pm \mathbb{E}[\tilde{M}_k] = \\
&\leq |\mathbb{E}[M_{n,k}] - \mathbb{E}[\tilde{M}_k]| + |\tilde{M}_k - \mathbb{E}[\tilde{M}_k]| + |\mathbb{E}[M_{n,k}] - M_{n,k}| \leq \\
&\leq \frac{k^2 + k}{n^2 - nk} + \sqrt{2log(\frac{4}{\delta}) \frac{(k+1)^2}{n}} + \sqrt{\frac{1}{nb_k} log(\frac{8}{\delta})} = \\
&= \sqrt{\frac{1}{nb_k} log(\frac{8}{\delta})} + (k+1)\Big[\frac{k}{n^2 + nk} + \sqrt{\frac{2}{n} log(\frac{4}{\delta})}\Big]
\end{aligned}$$

We can inflate a little bit the second inequality of the theorem, so we can isolate the constant C_k that appears in Bubeck's algorithm:

$$M_k \leq \tilde{M}_k + \sqrt{\frac{1}{nb_k} log(\frac{4}{\delta})} + (k+1)\Big[\frac{k}{n^2+nk} + \sqrt{\frac{2}{n} log(\frac{2}{\delta})}\Big] \leq$$
$$\leq \tilde{M}_k + \frac{k^2+k}{n^2-nk} + \sqrt{\frac{1}{n} log(\frac{4}{\delta})}\Big[\sqrt{\frac{1}{b_k}} + \sqrt{k}(k+1)\Big]$$
$$C_k = \sqrt{\frac{1}{b_k}} + \sqrt{k}(k+1)$$

Similarly, we find a term $f(n,k) = \frac{k^2+k}{n^2-nk}$. This function attains its maximum value at $k = \frac{n}{2}$. One can notice that is similar to $-\frac{1}{n}$ of *Proposition 1* of [1], and because $f(n, k = \frac{n}{2})$ converges to zero as $n \to \infty$, we do not have to place it in the algorithm for the Multi-armed bandit problem.

Algorithm 3 Good-Turing Algorithm for frequencies $k > 0$

for j in $1 : J$ **do**
  Sample $\frac{n_{init}}{J}$ units from each population
end for

for i in 1:additional sample **do**
  Find GT_{i-1} = {units with frequency $k-1$ in the joint sample}
  for j in 1:J **do**
    Find $U_{j,i-1}$ = {units $\in GT_{i-1}$ and sampled from $j-th$ population }
    Compute $r_j = \frac{|U_{j,i-1}|}{n_{j,i-1}} + C_k\sqrt{\frac{log[4(n_{init}+i)]}{n_{j,i-1}}}$
  end for
  Compute $j^* = argmax\{r_j : j = 1, 2, ..., J\}$
  Sample next observation from population j^*
end for

With respect to the estimation of the missing mass, that is when $k = 0$, using the inequality of *Theorem 16* in [9] for $|\mathbb{E}[M_{n,r}] - M_{n,r}|$ rather than the one of *Theorem 5* in [11] one gets with probability at least of $1-\delta$ the upper confidence bound of *Proposition 1* in [1]:

$$M_k \leq \tilde{M}_k + (1+\sqrt{2})\sqrt{\frac{log(\frac{4}{\delta})}{n}}$$

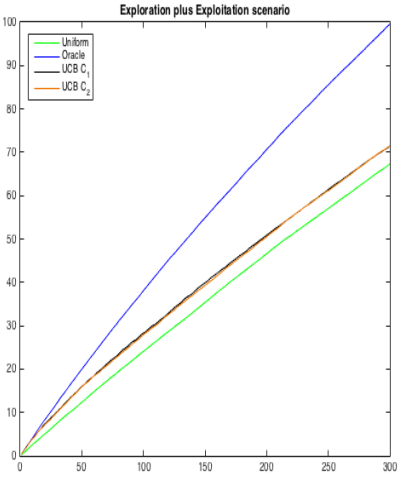
Where the tuning constant seems to be $C_0 = (1 + \sqrt{2})$. However, the authors use $C_0 = (1 + \sqrt{2})\sqrt{3}$ that comes from the regret analysis. Otherwise, one can apply the general inequality, in this case we can pick:

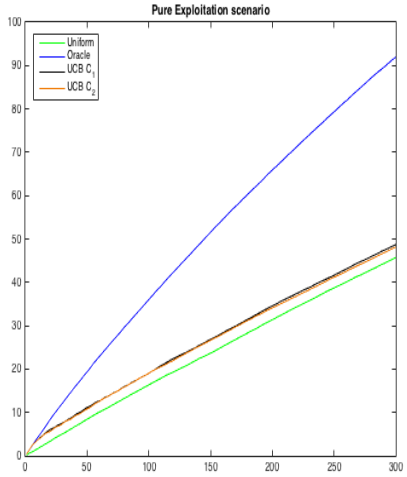
$$C_0' = \sqrt{8e}$$

.

To conclude the section, I analyse the performances of the UCB algorithm in mazimazing the number of different species discovered. To do so, I use the constants C_0 and C_0'. The simulations have been done with the three settings introduced in **Section 2.4**: Pure exploration, Exploration plus Exploitation and Pure exploitation. We can check from the plots (the black line is for C_0 and the orange one for C_0') that both the algorithms produce an improvement from the Uniform strategy. Equivalently, as we expected, the UCB with constat C_0 works slightly better since the inequality of [9] for $|\mathbb{E}[M_{n,r}] - M_{n,r}|$ is tighter that the one in *Theorem 5* of [11]. Indeed, remember that the constant C_0 of Bubeck is chosen to minimize the expected regret.

The minimization of the expected regret for the upper confidence bound of **Theorem 3.1** is not the purpuse of this work, but it would certaintly be of interest to carry on a regret analysis for the GT estimator $\tilde{M}_k$ as done in [1] for $\tilde{M}_0$.





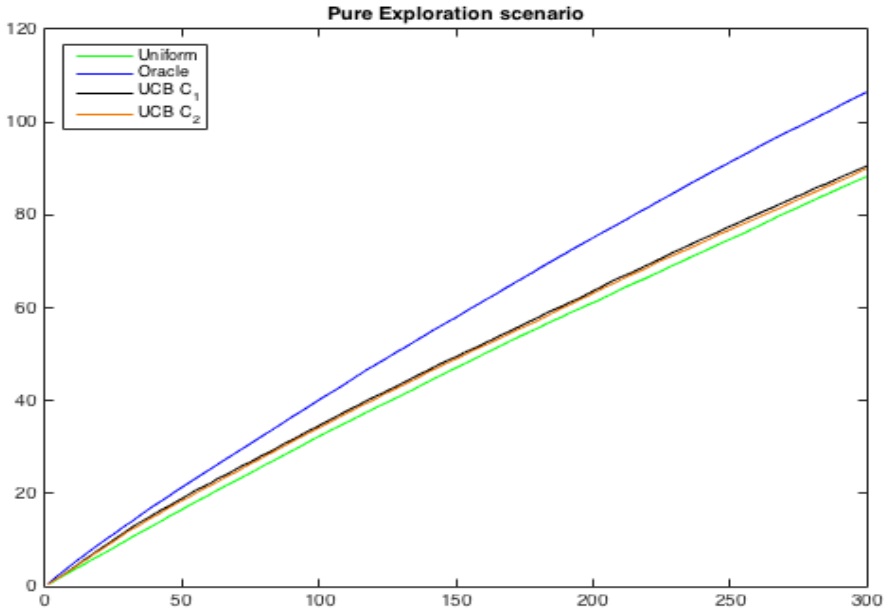


Figure 2: Comparison of UCB strategies with costants $1+\sqrt{2})\sqrt{3}$ in black and with $C_0' = \sqrt{8e}$ in orange for the three scenarios of Exploitation plus Exploration, Pure Exploitation and Pure Exploration.

3.2 Generalized HPY-TS

We now move to the generalization of the HPY-TS strategy for the case where we want to target species already observed. Recall that $\mathbf{Y_n} = (\mathbf{Y}_{n_{1..}}, \mathbf{Y}_{n_{2..}}, ..., \mathbf{Y}_{n_{J..}})$ is the joint sample from the J populations. Similarly to what we have done for the missing mass $k = 0$, we define the set of interest A_k of species that have frequency k in the overall sample $\mathbf{Y_n}$:

$$A_k = \{\theta \in \{\theta_1^*, \theta_2^*, ..., \theta_K^*\} : card(\{(i,j) : \theta_{i,j} = \theta\}) = k\}$$

As before, we are interested in the posterior distribution of $(P_1(A_k), P_2(A_k), ..., P_J(A_k))|\mathbf{Y_n}$, where the conditioning on $\sigma_j, \theta_j, \alpha, \gamma, H$ is also omitted. Suppose that there are $\bar{k}$ dishes served with frequency k and they coincide with $\theta_1^{**}, \theta_2^{**}, ..., \theta_{\bar{k}}^{**}$, unless a relabeling of the distinct dishes is needed. The following result is provided in the online notes of [2] that are available online.

Proposition 3.4. *Let $\mathbf{Y_n}$ denote the joined sample from a HPY process and let A_k be defined as above. Then, $(P_1(A_k), P_2(A_k), ..., P_J(A_k))|\mathbf{Y_n}$ admits the following*

multivariate density:

$$f_{(P_1(A_k),..,P_J(A_k))|\mathbf{Y_n}}(p_1,...,p_J) = \int_0^1 \prod_{j=1}^{J} f_j(p_j|\mathbf{Y_n},\beta_0) f_0(\beta_0|\mathbf{Y_n}) d\beta_0 \qquad (33)$$

where:

$$\begin{aligned} &f_j(p_j|\mathbf{Y_n},\beta_0) = beta(p_j|a_j,b_j) \\ &a_j = (\theta_j + m_{j.}\sigma_j)\beta_0 + N_{jk} - \sigma_j M_{jr} \\ &b_j = (\theta_j + m_{j.}\sigma_j)(1-\beta_0) + n_{j..} - N_{jk} - \sigma_j(m_{j.} - M_{jr}) \end{aligned}$$

$$f_0(\beta_0|\mathbf{Y_n}) = beta(\beta_0|M_{.k} - k\alpha, \gamma + k\alpha + m_{..} - M_{.k})$$

Having defined

$$N_{jk} = \sum_{i=1}^{\bar{k}} n_{j.i} \quad \textit{and} \quad M_{jk} = \sum_{i=1}^{\bar{k}} m_{ji.}$$

As for the missing mass, it is possible to derive the posterior mean $\mathbb{E}(P_j(A_k)|\mathbf{Y_n})$. Anyway, following the prescription of the Thompson-sampling strategy, we will draw a sample according to $f_j(p_j|\mathbf{Y_n},\beta_0)$ and $f_0(\beta_0|\mathbf{Y_n})$. The strategy is summarized in the algorithm present in the following page.

3.3 Discussion and Simulations

Let us concentrate on the frequency $k = 1$, meaning that we target species observed once in the sample. In this case, the winning arms will be those with the highest probability of sampling one of the species belonging to A_1. What changes with respect to the maximization of unobserved species is that the posterior distributions of the rewards are not non-increasing functions of the number of draws anymore. In fact, as time moves forward, we may sample new species, so that the cardinality of the set A_1 increases.

For this reason, it is arduous to interpret the object of maximization in this framework. When $k = 1$, one can think of sampling the highest number of units with frequency $k > 1$. However, when $k > 1$ we fail to keep this interpretation. One could think of running out of the units with a certain frequency, however we won't sample all the species in A_k as this set continues to attract new units.

Algorithm 4 TS-HPY Algorithm for frequencies $k = 1, 2, .$

for j in $1 : J$ **do**

Sample $\frac{n_{init}}{J}$ units from each population

end for

Calculate table counts and estimate the HPY hyperparameters

for i in 1:additional sample **do**

Draw $\beta_0 \sim beta(M_{.k} - k\alpha, \gamma + k\alpha + m_{..} - M_{.k})$, where:

$$N_{jk} = \sum_{i=1}^{\bar{k}} n_{j.i} \quad \textit{and} \quad M_{jk} = \sum_{i=1}^{\bar{k}} m_{ji.}$$

for j in 1:J **do**

Draw $p_j \sim beta(a_j, b_j)$, where:

$$a_j = (\theta_j + m_{j.}\sigma_j)\beta_0 + N_{jk} - \sigma_j M_{jr}$$
$$b_j = (\theta_j + m_{j.}\sigma_j)(1 - \beta_0) + n_{j..} - N_{jk} - \sigma_j(m_{j.} - M_{jr})$$

end for

Compute $j^* = argmax\{p_j : j = 1, 2, ..., J\}$

Sample next observation from population j^*

Update table counts and estimates of the HPY hyperparameters

end for

A much simpler formalization consists in considering the units with a certain frequency k in the initial sample I and trying to reobserve them. Anyway, both HPY-TS and the UCB algorithms are suboptimal for this purpose as they are maximizing the probability of observing individuals belonging to a set which is not I.
In the following plot we compare HPY-TS, Generalized UCB, Oracle and Uniform for frequencies $k = 1$ in the Pure Exploration and Exploration plus Exploitation settings when trying to reobserve units in the initial sample. For this purpose, we set an initial sample equal to 100 units in each population and run the algorithm for 150 additional time steps. As one can check, the HPY-TS achieves higher discoveries than the other strategies, whereas the Generalized UCB works better than Uniform in the Exploration plus Exploitation setting.

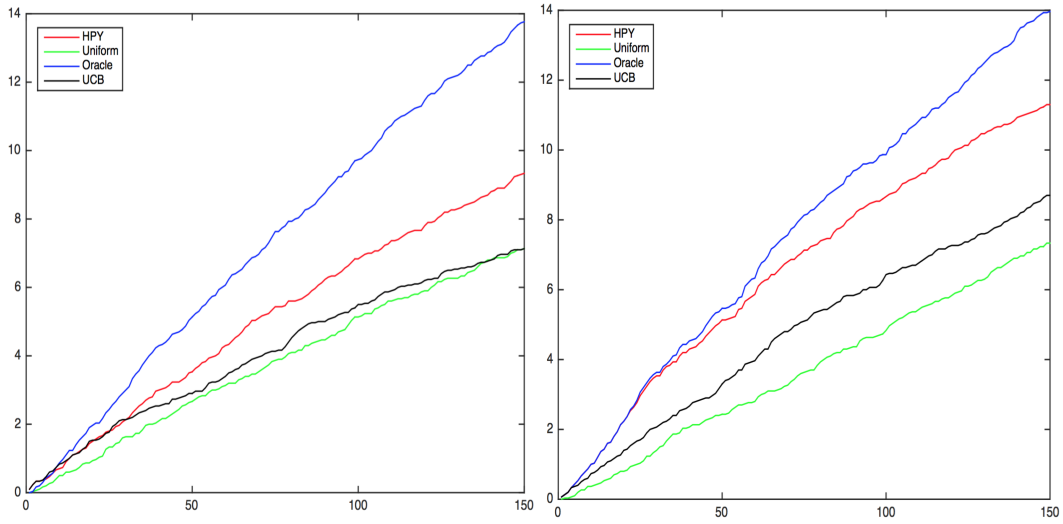


Figure 3: Comparison between HPY-TS, Generalized UCB, Oracle and Uniform for frequencies $k = 1$ in the Pure Exploration (left) and Exploration plus Exploitation settings (right)

4 Application to Political Polls

In this section I provide an application of the UCB and HPY-TS to political polls. Recall that our algorithm chooses the population to explore based on which ones could provide the greatest source of etherogeneity. This application makes sense because electoral polls should incorporate diversified opinions as much as possi-

ble. The data are taken from the monthly survey done by the Pew Research Center, a non partisan American "fact tank". They can be downloaded at Pew Research Center-datasets. Each monthly survey is approximately conducted on 2000 respondents. The questionnaire contains both specific questions useful for statistical analysis (age, education, sex, etc) but also questions regarding trends and attitudes on general topics and on incoming national elections. The data are collected using the Random Digit Dialing method (RDD). That consists in "selecting people for involvement in telephone statistical surveys by generating telephone numbers at random" (Wikipedia). Still, the RDD allows to choose the country code and the area code, that in our setting means the opportunity to select the population from which we want to sample a unit. After choosing the population, which correspond to a geographical area, the individual is randomly sampled. Actually, just half of the sampling is done calling landline phones, whereas the other half targets the mobile phones. Nevertheless, in the US cellphones have an area code. Even though people can move from one state to another, the difference between the expected location using the area code the actual location from the Zip code is not significant, as one can check from the data.

Nowadays, the trend of collecting data for polls using new methods such as online questionairres is on the rise. We refer to this kind of surveys, either from internet panels or from self-selected samples, as non-probability samples, due to the lack of randomness in the selection of units. Indeed, with non-probability samples we cannot calculate *"how likely the findings from the sample accurately represent the full population. That is, we can calculate the margin of sampling error, which is basically the price we pay for not interviewing every member of the population" (Sampling Methods for Political Polling - AAPOR)*. On the other hand, the number of probabilistic surveys is increasing as well, and the algorithm we propose could be well-employed in this setting. Another concern may be to the decline in response rates in RDD sampling, but fortunately there is not *much evidence of a crisis in the results pollsters were obtaining. (The State Of The Polls, 2016 - Nate Silver)*.

To convince ourselves about the importance of probabilistic samples and representativeness of the pool, the case of Literary Digest is symbolic. During the 1936 election campaign between the Democratic candidate Franklin Delano Roosevelt and the Republican candidate Alf Landon, the weekly magazine Literary Digest conducted a mail-in survey on over two million respondents. The journal inaccurately predicted a landslide victory for Republican candidate Alf Landon, who actually

lost in every state except for Main and Vermont. The main issue with the poll was the high selection bias, as the respondents were mostly auto and telephone owners, that over-represented middle and upper-class voters, excluding the electoral base of the incumbent Franklin Roosevelt. Since then, methods to correct for the selection bias have evolved and refined, making it possible to borrow information from both probabilistic and non-probabilistic samples. Some interesting examples are the papers [3] and [19]. The latter is particularly intriguing as it proposes a model to adjust via multilevel regression and poststratification a highly non-representative dataset obtained through an Xbox intention survey, and afterwards exploits the data to predict the 2012 US national elections. A review of biases from several samples' types can be found in [14].
As pointed out by the Pew Research center, *"For some surveys, it is important to ensure that there are enough members of a certain subgroup in the population so that more reliable estimates can be reported for that group. To do this, we oversample members of the subgroup [..]. However, this approach often increases costs [..].* It becomes clear how our algorithm could be helpful, as it could improve the number of individuals coming from different subgroups.
On the other hand, one could argue that in US it doesn't matter who wins the popular vote but who wins in the electoral college. This is true, but also in these questionnaires focused on the incoming elections, there are a lot of questions designed for the understanding of general topics and current attitudes. Besides, 2000 units isn't enough for the estimation on every singular state. Finally, the election polls are only a small part of the totality of surveys conducted, and this algorithm could be applied to have heterogeneity across individuals.

4.1 Crossclassification of species

We now move to a precise description of how I conducted the analysis on the data from the Pew Research center. To crossclassify individuals as different species, I use the variables Age (18-24,25-34,35-44,45-54,55-65,65-99), Education (High school graduate or less, College but no degree, Degree and postgraduate education), Sex (Male and Female) and Race (White, Black or African-American, Asian or Asian-American, Hispanic/Latino, Other). Moreover, in a second analysis, I will consider as species the Unlikely voters and the Undecided voters regardless of their age, education, sex and race, but I will explain this part later.

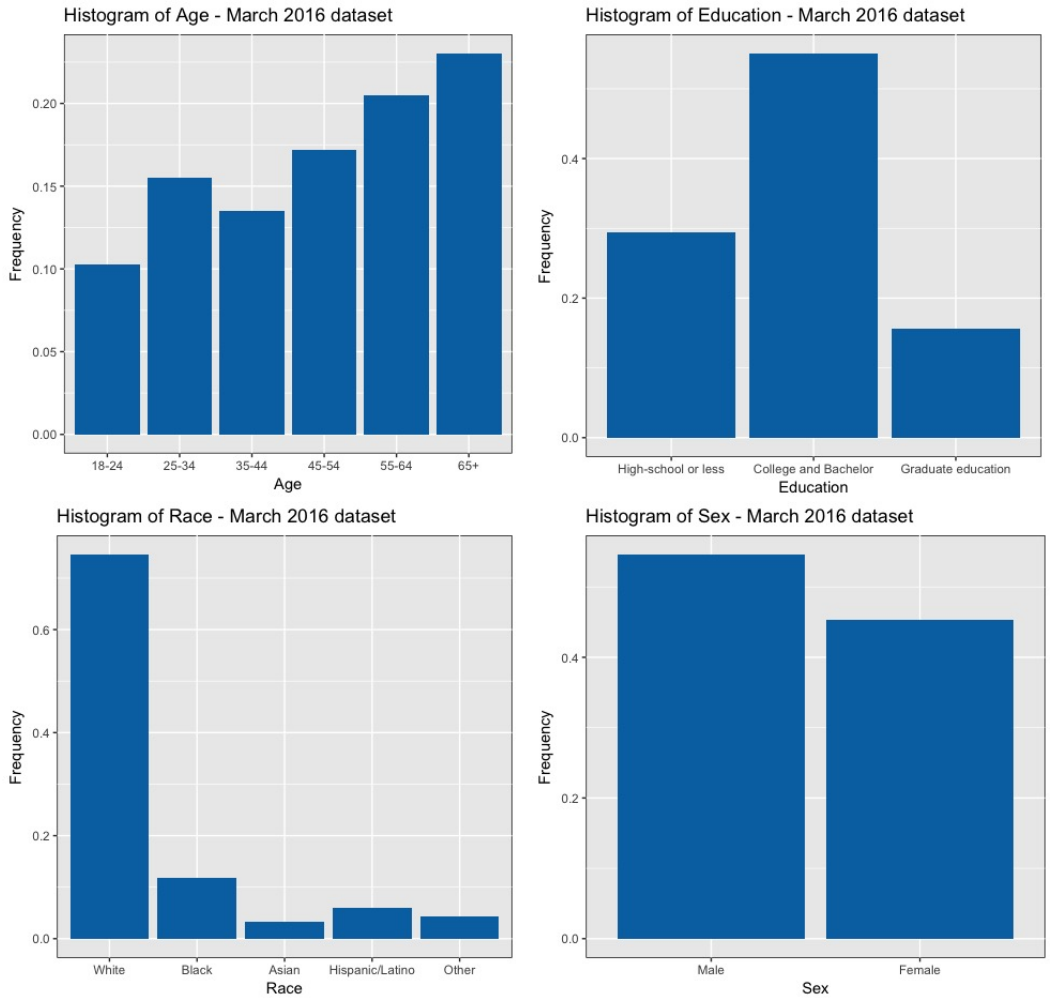


Figure 4: Histograms of the variables Age, Education, Race and Sex for March 2016 Dataset of Pew Research Center

I crossclassified the individuals in 10 populations using the division provided by OMB (Office of Management and Budget) in April, 1974:

1. Connecticut, Maine, Massachusetts, New Hampshire, Rhode Island, Vermont
2. New Jersey, New York, Puerto Rico, US Virgin Islands
3. Delaware, District of Columbia, Maryland, Pennsylvania, Virginia, West Virginia
4. Alabama, Florida, Georgia, Kentucky, Mississippi, North Carolina, South Carolina, Tennessee

5. Illinois, Indiana, Michigan, Minnesota, Ohio, Wisconsin
6. Arkansas, Louisiana, New Mexico, Oklahoma, Texas
7. Iowa, Kansas, Missouri, Nebraska
8. Colorado, Montana, North Dakota, South Dakota, Utah, Wyoming
9. Arizona, California, Hawaii, Nevada, American Samoa, Guam, Northern Mariana Islands, Trust Territory of the Pacific Islands
10. Alaska, Idaho, Oregon, Washington

4.1.1 The Unlikely Voter

When doing electoral polling, we are not interested in knowing the favourite candidate of people that are unlikely to vote. For this reason, pollsters ask some specific questions that help predicting weather or not the individual will vote. Then, to each respondent they assign a score and exclude a percentage based on the expected turnout (calculated using past election data and current trends).
In the questionnaire, there are two questions that help screening the likely voters. The first one (REG) is about being registred to vote, whereas the second question (OFTVOTE) is asked only to respondents that answer REG=1.

(REG) Which of these statements best describes you?:

1. Are you ABSOLUTELY CERTAIN that you are registered to vote at your current address
2. Are you PROBABLY registered, but there is a chance your registration has lapsed
3. Are you NOT registered to vote at your current address
4. Don't know/Refused

(OFTVOTE) How often would you say you vote?

1. Always
2. Nearly always

3. Part of the time
4. Seldom
5. Never Vote, Don't Know, Other (people answering REG $\neq$ 1)

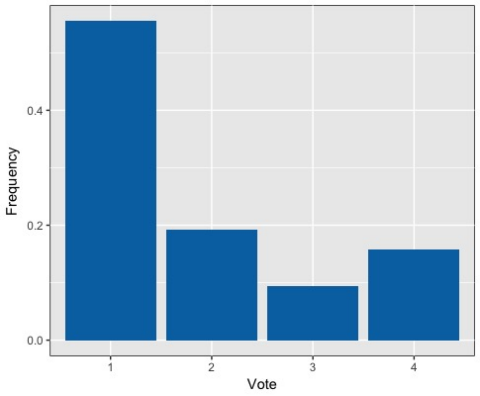


Figure 5: Variable OFTVOTE March 2016

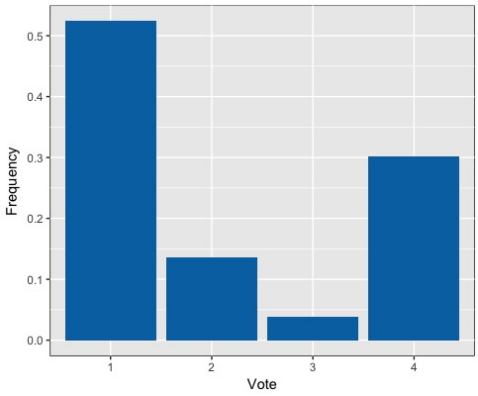


Figure 6: Variable OFTVOTE June 2016

These two questions clearly produce an ordinal list that tells us which respondents have an higher probability of voting. For instance, it is more likely that someone responding REG=1 and OFTVOTE=2 goes to the voting station rather than another person who answeered REG=2. At this stage we are interested in classifying Unlikely Voters as a single species, since we are not interested in "discovering" them. As already said, usually pollster consider a weigthed mean of latest elections to estimate the turnout. In this work, I will consider as likely voters the respondents that answer OFTVOTE=1,2 and OFTVOTE=1 in two different analysis. The frequency of the expected people voting using both who answered OFTVOTE=1,2 is higher (around 75 per cent) than a reasonable expectation of an high turnout (around 60 per cent). Whereas, using only OFTVOTE=1, it is lower (around 50 per cent). However, one should also take into account the selection bias, as people that take time to answer the questionnaire are usually more interested in politics than the ones that don't. Nevertheless, I consider these two cases as an upper and a lower bound of the percentage of people classified as likely voters.
Summarizing, likely voters will be classified as different species using the attributes age, sex, race and education. On the other hand, the unlikely voters will be included as a single species.

4.1.2 The Undecided Voter

Undecided Voters is another class of people that has to be considered all together. The main reason is that from them you do not obtain any information useful for the prediction of the winning canditate in the next election. On the other hand it is useful to identify the proportion of undecided voters because they increase the variability in polls estimation.
For example, during the latest elections, two weeks before the voting day "about 15 percent of the electorate wasn't yet committed to Clinton or Trump, compared to just 5 percent who weren't committed to President Obama or Mitt Romney at this point in 2012" (Nate Silver-"Election Update: Where Are The Undecided Voters?"). This was the reason why the electoral voting model of FiveFirtyEight was giving a greater probability of winning to Donald Trump with respect to other models. In addition, undecided voters are clearly not uniformly distributed across states.
From our practical point of view, we do not make a distinction between undecided voters and third party voters in the majority of state since third party voters have often a weak commitment to their candidate (Nate Silver-"Election Update: Where Are The Undecided Voters?"). To screen for undecided voters, I considered people answering *"Don't know/Refused"* to the following question.
(PARTYLN) As of today do you lean more to the Republican Party or more to the Democratic Party?

1. Republican
2. Democrat
3. Don't know/Refused

4.2 Simulations March 2016 dataset

In the first simulation I classified the individuals as specified in the previous sections without accounting for undecided voters and unlikely voters. The dataset considered is the 2016 montly survey of the Pew Research center. I first sampled 50 units in every population, and the let the different strategies choose the population for 150 additional samples, while the algorithm is run in total 40 times. The results of the discovered species are summarized in the following plot:

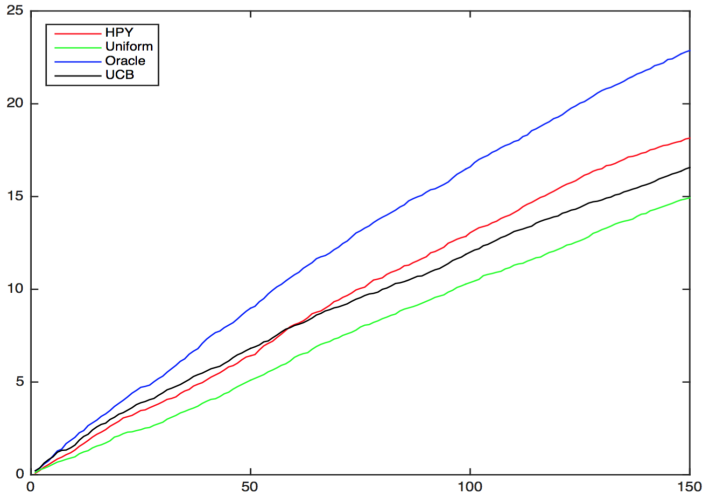


Figure 7: Results of the simulations using March 2016 dataset when 50 units are sampled at the beginning and 150 units additionally when the goal is discovering the highest number of different species.

We may compare different strategies considering the average number each strategy sampled from every population, the results the 2016 montly survey of the Pew Research center are shown in the table below:

Table 1: Average number of times every population has been sampled from the strategies Oracle, TS-HPY, Good-Turing and Uniform.

Population	Oracle	TS-HPY	Good-Turing	Uniform
1	0	4.4	11.36	15.2
2	28.5	23.5	17.15	14.9
3	0	12.9	17.52	15.21
4	0	19.7	16.8	14.63
5	0	25	14.9	15.35
6	37.7	19.05	16.3	14.56
7	0	7.9	12.05	14.72
8	0	4.2	11.11	15.66
9	83.8	23.1	19.4	14.88
10	0	10.23	13.5	14.87

The three population sampled by the Oracle strategy they have an high number of unique species divided by total species and the estimated Zipf coefficients are lower that the median coefficient, and they are significant.
As one can see, the TS-HPY performs better that the GT strategy since it avoids sampling from unprofitable populations, such as the 1^{st}, the 7^{th} and the 8^{th}, while exploring the profitable arms. Finally, the GT strategy and the Uniform ones performs similarly.
In the second simulation I classified as likely voters early half of the respondents at the questionnaire, that is only people answering OFTVOTE=1, while the other half made a single species of unlikely voters. I also classified the undecided voters as a single species and increased the number of additional samples from 150 to 300.
As one can see from the above plot, both the HPY-TS and the GT strategy perform slightly worse than before. One of the reasons is that the number of unique species in the populations dropped from 238 to around 180. Despite the increased difficulty on observing new species, the TS-HPY manages to outperform the GT and Uniform strategies.

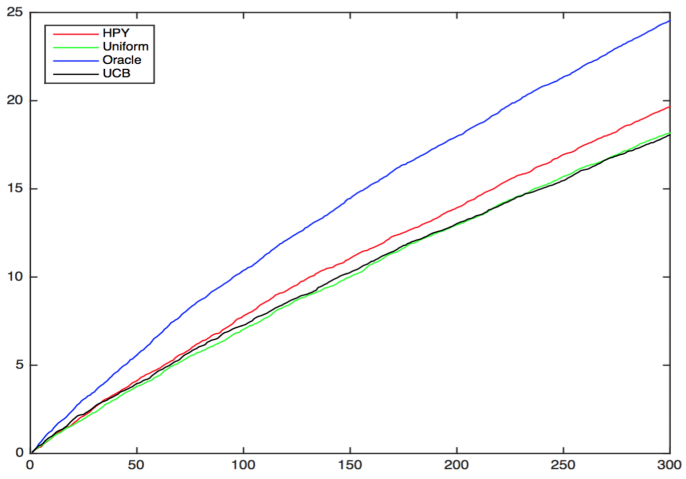


Figure 8: Results of the simulations using March 2016, considering Unlikely and Undecided voters, when the goal is discovering the highest number of different species.

5 Appendix A - Algorithms

Algorithm 5 UCB for the missing mass

for j in $1:J$ **do**
  Sample $\frac{n_{init}}{J}$ units from each population
end for
for i in 1:additional sample **do**
  Find GT_{i-1} = {units with frequency $r = 1$ in joint sample}
  for j in 1:J **do**
    Find $U_{j,i-1}$ = {units $\in GT_{i-1}$ and sampled from $j-th$ population }
    Compute $r_j = \frac{|U_{j,i-1}|}{n_{j,i-1}} + C\sqrt{\frac{log[4(n_{init}+i)]}{n_{j,i-1}}}$
  end for
  Compute $j^* = argmax\{r_j : j = 1, 2, ..., J\}$
  Sample next observation from population j^*
end for

Algorithm 6 Oracle algorithm for the missing mass

for j in $1:J$ **do**
  Sample $\frac{n_{init}}{J}$ units from each population
end for
for i in 1:additional sample **do**
  Let A be the set of unique units sampled
  for j in 1:J **do**
    Let $f_{1,j}, f_{2,j}, ..f_{|A|,j}$ be the frequency of A's elements in the j-th population
    Calculate the missing mass $h_j = \sum_{k=1}^{|A|}(1 - f_{k,j})$
  end for
  Compute $j^* = argmax\{h_j : j = 1, 2, ..., J\}$
  Sample next observation from population j^*
end for

Algorithm 7 TS-HDP for the missing mass

for j in $1 : J$ **do**
  Sample $\frac{n_{init}}{J}$ units from each population
end for
Calculate table counts and estimate the HPY hyperparameters

for i in 1:additional sample **do**
  Draw $\beta_0 \sim beta(\gamma, m_{..})$
  for j in 1:J **do**
    Draw $p_j \sim beta(\beta_0\theta_j, \theta_j(1-\beta_0) + n_{j..})$
  end for
  Compute $j^* = argmax\{p_j : j = 1, 2, ..., J\}$
  Sample next observation from population j^*
  Update table counts and estimates of the HPY hyperparameters
end for

6 Appendix B - Codes for Bootstrap Filter

```
% Function that updates the Hyperparameters using the
   particle filter of Liu and West.
function [ alpha, d ,gamma ,nu M_iperparametri_new]=Filter_
   iperparametri(mjk,m_j_dot,m_dd,m_dot_k,n_j_dot_k,J,nn,
   bigK,N_iter,M_iperparametri_old)

%smoothing parameter
h=1/N_iter;

%point A) of the algorithm
a=sqrt(1-h^2);
E_parametri=mean(M_iperparametri_old);
cov_parametri=cov(M_iperparametri_old);
medie_mx=a*M_iperparametri_old+(1-a)*ones(N_iter,1)*E_
   parametri;

%Function that computes the weights at point B.1) of the
   algorithm
g=pesi_filter(mjk,m_j_dot,m_dd,m_dot_k,n_j_dot_k,J,nn,medie_
```

```
    mx,bigK,N_iter);
g_cum=cumsum(g);

%Function for points B.2) and B.3)
M_iperparametri_new=zeros(N_iter,2*J+2);
[M_iperparametri_new_unnormalized omega]=pesi_filter_new(mjk
    ,m_j_dot,m_dd,...
     m_dot_k,n_j_dot_k,J,nn,bigK,N_iter,g,g_cum,h,medie_mx,
        cov_parametri);

%Resample to obtaine a set of parameters with equal weights,
     point C)
omega_cum=cumsum(omega);
for ii=1:N_iter
     U=unifrnd(0,1);
     ind=find(U<omega_cum);
     ind=ind(1);
     M_iperparametri_new(ii,:)=M_iperparametri_new_
        unnormalized(ind,:);
end

%Find a point estimates of the hyperparameters, such as the
    mean
alpha=mean(M_iperparametri_new(:,1:J));
d=mean(M_iperparametri_new(:,(J+1):(2*J)));
gamma=mean(M_iperparametri_new(:,2*J+1));
nu=mean(M_iperparametri_new(:,2*J+2));
```

```
%Function that computes the weights at point B.1) of the
    algorithm
function p=pesi_filter(mjk,m_j_dot,m_dd,m_dot_k,n_j_dot_k,J,
    nn,medie_mx,bigK,N_iter)

vec=1:(bigK-1);
```

```
logp=zeros(1,N_iter);

for jj=1:N_iter
    alpha=medie_mx(jj,1:J);
    d=medie_mx(jj,(J+1):(2*J));
    gamma=medie_mx(jj,2*J+1);
    nu=medie_mx(jj,2*J+2);
Phi=0;

%computation of the EPPF for restaurant j
for j=1:J

    vec_j=1:(m_j_dot(j)-1);

    %Computation of the maximum of n_j.k so that I can use
       it to calculate the generalized factorial coefficient
    max_nj=max(n_j_dot_k(j,:));
    max_mj=max(mjk(j,:));
    LogC=generalized_factorial(max_nj,max_mj,d(j));

    %big K is the number of unique values
    vec_loggfc=zeros(1,bigK);

    for ii=1:bigK
        vec_loggfc(ii)=LogC(n_j_dot_k(j,ii)+1,mjk(j,ii)+1);
    end

    %summation of the log-EPPF
    Phi_j=sum(log(alpha(j)+vec_j*d(j)))-gammaln(alpha(j)+nn(
       j))+gammaln(alpha(j)+1)+...
        sum(vec_loggfc)-m_j_dot(j)*log(d(j));
    Phi=Phi+Phi_j;

end
%EPPF for the tables (chinese restaurant franchise)
logp(jj)=Phi+sum(log(gamma+nu*vec))-gammaln(gamma+m_dd)+
   gammaln(gamma+1)+...
```

```
    sum(gammaln(m_dot_k-nu))-bigK*gammaln(1-nu);
end

%Normalization of the log-weights and computation of
   exponential to get
%probabilities
M=ones(N_iter,1)*logp;
p=1./sum(exp(M-M'),2)';
```

```
%Function for points B.2) and B.3)
function [M_iperparametri_new_unnormalized omega]=pesi_
   filter_new(mjk,m_j_dot...
    ,m_dd,m_dot_k,n_j_dot_k,J,nn,bigK,N_iter,g,g_cum,h,medie
       _mx,cov_parametri)

p=zeros(1,N_iter);

%vector for the log weights of point B.3) in the algorithm
logp_new=zeros(1,N_iter);

%vector for the new MC sample of hyperparameters
M_iperparametri_new_unnormalized=zeros(N_iter,2*J+2);

vec=1:(bigK-1);

%Point B.2) of the algorithm
for jj=1:N_iter
    U=unifrnd(0,1);
    ind=find(U<g_cum);
    ind=ind(1);
    M_iperparametri_new_unnormalized(jj,:)=mvnrnd(medie_mx(
       ind,:),h^2*cov_parametri);
    alpha=M_iperparametri_new_unnormalized(jj,1:J);
    d=M_iperparametri_new_unnormalized(jj,(J+1):(2*J));
```

```
23      gamma=M_iperparametri_new_unnormalized(jj,2*J+1);
24      nu=M_iperparametri_new_unnormalized(jj,2*J+2);
25      Phi=0;
26
27      %Computation of the EPPF's as in the function pesi_
           filter, this time
28      %for point B.3) in the algorithm
29      for j=1:J
30          vec_j=1:(m_j_dot(j)-1);
31          max_nj=max(n_j_dot_k(j,:));
32          max_mj=max(mjk(j,:));
33          LogC=generalized_factorial(max_nj,max_mj,d(j));
34          vec_loggfc=zeros(1,bigK);
35          for ii=1:bigK
36              vec_loggfc(ii)=LogC(n_j_dot_k(j,ii)+1,mjk(j,ii)
                   +1);
37          end
38          Phi_j=sum(log(alpha(j)+vec_j*d(j)))-gammaln(alpha(j)
               +nn(j))+gammaln(alpha(j)+1)+...
39              sum(vec_loggfc)-m_j_dot(j)*log(d(j));
40          Phi=Phi+Phi_j;
41      end
42
43      logp_new(jj)=Phi+sum(log(gamma+nu*vec))-gammaln(gamma+m_
           dd)+gammaln(gamma+1)+...
44          sum(gammaln(m_dot_k-nu))-bigK*gammaln(1-nu);
45      p(jj)=g(ind);
46  end
47
48  %Normalization of weights
49  M=ones(N_iter,1)*logp_new;
50  p_new=1./sum(exp(M-M'),2)';
51
52  %Computation of the importance sample weights of point B.3)
53  omega=p_new./p;
54  omega=omega/sum(omega);
```

References

[1] BUBECK, S., ERNST, D. and GARIVIER, A. (2013). Optimal discovery with probabilistic expert advice: finite time analysis and macroscopic optimality. *Journal of Machine Learning Research*, **14**, 601-623.

[2] BATTISTON, M., FAVARO, S. and TEH, Y.W. (2016). Multi-armed bandit for species discovery: A Bayesian nonparametric approach. *Journal of the American Statistical Association*

[3] BARBER, M., MANN, C., MONSON Q. and PATTERSON, K. (2014). Online Polls and Registration-Based Sampling: A New Method for Pre-Election Polling. In *Political Analysis*

[4] FERGUSON, T.S. (1973). A Bayesian analysis of some nonparametric problems. *The Annals of Statistics,* **1**, 209-230.

[5] FONTENEAU-BELMUDES, F., ERNS, D., DRUET, P., PANCIATICI, P. and WEHENKEL, L. (2010). Consequence driven decomposition of large-scale power system security analysis. In *Proceedings of the 2010 IREP Symposium.*

[6] FONTENEAU-BELMUDES, F. (2012). Identification of Dangerous Contingencies for Large Scale Power System Security Assessment. *PhD thesis, University of Liege*

[7] GOOD, I.J. (1953). The population frequencies of species and the estimation of population parameters. *Biometrika*, **40**, 237–264.

[8] LIU, J. and WEST, M. (2001). Combined Parameter and State Estimation in Simulation-Based Filtering. In *Sequential Monte Carlo Methods in Practice* 197-223

[9] MCALLESTER, D. and ORTIZ, L. (2003). Concentration Inequalities for the Missing Mass and for Histogram Rule Error. *Journal of Machine Learning Research*, **4**, 895-911

[10] MCDIARMID, C. (1989). On the method of bounded differences. In *Surveys in combinatorics*, volume **141** of London textitMath. Soc. Lecture Note Ser., pages 148–188. Cambridge Univ. Press, Cambridge, 1989.

[11] OHANNESSIAN, M. I. and DAHLEH, M. A. (2012). Rare Probability Estimation under Regularly Varying Heavy Tails *Proceedings of the 25th Annual Conference on Learning Theory*

[12] PITMAN, J. (1996). Some developments of the Blackwell-MacQueen urn scheme. In *Statistics, Probability andGame Theory* (eds. T. S. Ferguson, L. S. Shapley and J. B. MacQueen), pp. 245–267. Institute of Mathematical Statistics, Hayward.

[13] PITMAN, J. and YOR, M. (1997). The two parameter Poisson-Dirichlet distribution derived from a stable subordinator. *Annals of Probabability*, **25**, 855-900.

[14] ROTHSCHILD, D. (2009). Comparing prediction markets, polls, and their biases. *Public Opinion Quarterly*, Vol. *73*, No. 5 2009, pp. 895–916

[15] SILVER, N. (2016, June 2). The State Of The Polls, 2016. *FiveThirtyEight*. Retrieved from https://fivethirtyeight.com/politics/elections/

[16] SILVER, N. (2016, October 25). Election Update: Where are the undecided voters?. *FiveThirtyEight*. Retrieved from https://fivethirtyeight.com/politics/elections/

[17] TEH, Y.W. and JORDAN, M. (2010). Hierarchical Bayesian Nonparametric Models with Applications. In *Bayesian Nonparametrics* (eds. N.L. Hjort, C. Holmes, P. Muller and S.G. Walker), pp. 158-207. Cambridge University Press, Cambridge.

[18] THOMPSON, W.R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, **25**(3-4), 285–294.

[19] WANG, W., Rothschild, D., GOEL, B. and GELMAN, A. (2014). Forecasting elections with non-representative polls. *International Journal of Forecasting*

[20] WEST, M. (1993a). Approximating posterior distribution by mixtures, *Journal of Royal Statistical Society* **55**: 409-422

[21] WEST, M. (1993b). Mixture models, Monte Carlo, Bayesian updating and dynamic models. In J. H. Newton (ed.), *Computing Science and Statistics: Proceedings of the 24th Symposium on the Interface*, Interface Foundations of North America, Fairfax Station, Virginia, pp. 325-333