

---

# How Much is Unseen Depends Chiefly on Information About the Seen

---

Seongmin Lee<sup>1</sup> Marcel Böhme<sup>1</sup>

## Abstract

It might seem counter-intuitive at first: We find that, *in expectation*, the proportion of data points in an unknown population—that belong to classes that *do not* appear in the training data—is almost entirely determined by the number  $f_k$  of classes that *do* appear in the training data the same number of times. While in theory we show that the difference of the induced estimator decays exponentially in the size of the sample, in practice the high variance prevents us from using it directly for an estimator of the sample coverage. However, our precise characterization of the dependency between  $f_k$ 's induces a large search space of different *representations* of the expected value, which can be deterministically instantiated as estimators. Hence, we turn to optimization and develop a genetic algorithm that, given only the sample, searches for an estimator with minimal mean-squared error (MSE). In our experiments, our genetic algorithm discovers estimators that have a substantially smaller MSE than the state-of-the-art Good-Turing estimator. This holds for over 96% of runs when there are at least as many samples as classes. Our estimators' MSE is roughly 80% of the Good-Turing estimator's.

## 1. Introduction

Suppose, we are drawing balls with replacement from an urn with colored balls. What is the proportion of balls in that urn that have a color not observed in the sample; or equivalently, what is the probability that the next ball has a previously unobserved color? What is the distribution of rarely observed colors in that urn? These questions represent a fundamental problem in machine learning: How can we extrapolate from properties of the training data to properties of the unseen, underlying distribution of the data?

---

<sup>1</sup>Max-Planck Institute for Security and Privacy (MPI-SP), Germany. Correspondence to: Seongmin Lee <seongmin.lee@mpi-sp.org>, Marcel Böhme <marcel.boehme@mpi-sp.org>.

## 1.1. Background

Consider a *multinomial distribution*  $p = \langle p_1, \dots, p_S \rangle$  over a support set  $\mathcal{X}$  where support size  $S = |\mathcal{X}|$  and probability values are *unknown*. Let  $X^n = \langle X_1, \dots, X_n \rangle$  be a set of independent and identically distributed random variables representing the sequence of elements observed in  $n$  samples from  $p$ . Let  $N_x$  be the number of times element  $x \in \mathcal{X}$  is observed in the sample  $X^n$ . For  $k : 0 \leq k \leq n$ , let  $\Phi_k$  be the number of elements appearing exactly  $k$  times in  $X^n$ .

$$N_x = \sum_{i=1}^n \mathbb{1}(X_i = x) \quad \text{and} \quad \Phi_k = \sum_{x \in \mathcal{X}} \mathbb{1}(N_x = k)$$

Let  $f_k(n)$  be the expected value of  $\Phi_k$  (Good, 1953), i.e.,

$$f_k(n) = \binom{n}{k} \sum_{x \in \mathcal{X}} p_x^k (1 - p_x)^{n-k} = \mathbb{E}[\Phi_k]$$

**Estimating the multinomial.** Suppose, we want to estimate  $p$ . We cannot expect all elements to exist in  $X^n$ . While the empirical estimator  $\hat{p}_x^{\text{Emp}} = N_x/n$  is generally unbiased,  $\hat{p}_x^{\text{Emp}}$  distributes the entire probability mass only over the observed elements. This leaves a “missing probability mass” over the unobserved elements. In particular,  $\hat{p}_x^{\text{Emp}}$  given that  $N_x > 0$  overestimates  $p_x$ , i.e., for observed elements

$$\mathbb{E} \left[ \frac{N_x}{n} \mid N_x > 0 \right] = \frac{p_x}{1 - (1 - p_x)^n}.$$

We notice that the bias increases as  $p_x$  decreases. Bias is maximized for the rarest observed element.

**Missing mass, realizability, and natural estimation.** Good and Turing (GT) (Good, 1953) discovered that the expected value of the probability  $M_k = \sum_{x \in \mathcal{X}} p_x \mathbb{1}(N_x = k)$  that the  $(n+1)$ -th observation  $X_{n+1}$  is an element that has been observed exactly  $k$  times in  $X^n$  (incl.  $k = 0$ ) is a function of the expected number of colors  $f_{k+1}(n+1)$  that will be observed  $k+1$  times in an enlarged sample  $X^n \cup X_{n+1}$ .

$$\mathbb{E}[M_k] = \frac{k+1}{n+1} f_{k+1}(n+1). \quad (1)$$

We also call  $M_k$  as *total probability mass* over the elements that have been observed exactly  $k$  times. Since our sample  $X^n$  is only of size  $n$ , GT suggested to estimate  $f_{k+1}(n+1)$  using  $\Phi_{k+1}$ . Concretely,  $\hat{M}_k^G = \frac{k+1}{n} \Phi_{k+1}$ .

For  $k = 0$ ,  $M_0$  gives the “missing” (probability) mass over the elements not in the sample. In genetics and biostatistics, the complement  $1 - M_0$  measures *sample coverage*, i.e., the proportion of individuals in the population belonging to a species *not* observed in the sample (Chao & Jost, 2012). In the context of supervised machine learning, assuming the training data is a random sample, the sample coverage of the training data gives the proportion of all data (seen or unseen) with labels not observed in the training data.

Using an estimate  $\hat{M}_k$  of  $M_k$ , we estimate the probability  $p_x$  of an element  $x \in \mathcal{X}$  that appears  $k$  times as  $\hat{M}_k/\Phi_k$ . The estimation of  $p$  in the presence of unseen elements  $x \notin X^n$  is a *fundamental problem in machine learning* (Orlitsky et al., 2003; Orlitsky & Suresh, 2015; Painsky, 2022; Acharya et al., 2013; Hao & Li, 2020). For instance, in natural language processing the estimation of the probability of a given sequence of words occurring in a sentence is the main challenge of language models, particularly in the presence of sequences that appear in the training data rarely or not at all. Different smoothing and backoff techniques have been developed to tackle this data sparsity challenge.

A *natural estimator* of  $p_x$  assigns the same probability to all elements  $x$  appearing the same number of times in the sample  $X^n$  (Orlitsky & Suresh, 2015). For  $k > 0$ ,  $\hat{p}_x = M_k/\Phi_k$  gives the hypothetical<sup>1</sup> *best natural estimator* of  $p_x$  for every element  $x$  that has been observed  $k$  times.

**Bias of Good-Turing (GT).** In terms of bias, Juan and Lo (Juang & Lo, 1994) observe that the GT estimator  $\hat{M}_k^G = \frac{k+1}{n}\Phi_{k+1}$  is an unbiased estimate of  $M_k(X^{n-1})$ , i.e., where the  $n$ -th sample was *deleted* from  $X^n$  and find:

$$\begin{aligned} \left| \mathbb{E} \left[ \hat{M}_k^G - M_k \right] \right| &= \left| \mathbb{E} \left[ M_k(X^{n-1}) - M_k(X^n) \right] \right| \\ &\leq \frac{k+2}{n+1} = \mathcal{O} \left( \frac{1}{n} \right). \end{aligned}$$

**Convergence of GT.** McAllester and Schapire (McAllester & Schapire, 2000) analyzed the *convergence*. With high probability,

$$\begin{aligned} &|\hat{M}_k^G - M_k(n)| \\ &=_{\delta} \begin{cases} \mathcal{O}(1/\sqrt{n}) & \text{for } k = 0 \text{ (missing mass)} \\ \mathcal{O}(\log(n)/\sqrt{n}) & \text{for small } k \text{ (rare elements)} \\ \mathcal{O}(k/\sqrt{n}) & \text{for large } k \text{ (abundant elements)}. \end{cases} \end{aligned}$$

This result was improved by Drukh and Mansour (Drukh & Mansour, 2004) and more recently by Painsky (Painsky, 2022) who showed that GT estimator converges at a rate of  $\mathcal{O}(1/\sqrt{n})$  for all  $k$  based on worst-case mean squared error analysis.

<sup>1</sup>The best natural estimator is also called oracle-aided estimator for its knowledge about  $p_x$  (Orlitsky & Suresh, 2015) but cannot actually be used for estimation.

**Competitiveness of GT.** Using the Poisson approximation, Orlitsky and Suresh (Orlitsky & Suresh, 2015) showed that natural estimators from GT, i.e.,  $\hat{p}_x^G = \hat{M}_{N_x}^G/\Phi_{N_x}$ , performs close to the best natural estimator. Regret, the metric of the competitiveness of an estimator against the best natural estimator, is measured as KL divergence between the estimate  $\hat{p}$  and the actual distribution  $p$ ,  $D_{KL}(\hat{p}||p)$ . Orlitsky and Suresh also showed that finding the best natural estimator for  $p$  is same as finding the best estimator for  $M = \{M_k\}_{k=0}^n$ .

**Poisson approximation.** The Poisson approximation with parameter  $\lambda_x = p_x n$  has often been used to tackle a major challenge in the formal analysis of the missing mass and natural estimators (Orlitsky & Suresh, 2015; Orlitsky et al., 2016; Acharya et al., 2013; Efron & Thisted, 1976; Valiant & Valiant, 2016; Good, 1953; Good & Toulmin, 1956; Hao & Li, 2020). The challenge is the *dependencies between frequencies*  $N_x$  for different elements  $x \in \mathcal{X}$ . In this Poisson Product model, a continuous-time sampling scheme with  $S = |\mathcal{X}|$  independent Poisson distributions is considered where the frequency  $N_x$  of an element  $x$  is represented as a Poisson random variable with mean  $p_x n$ . In other words, in the Poisson approximation, the frequencies  $N_x$  are modelled as independent random variables. Consequently, GT estimator is an unbiased estimator for the Poisson Product model (Orlitsky et al., 2016), yet it is biased in the multinomial distribution (Juang & Lo, 1994). Hence, we tackle the dependencies between frequencies analytically, without approximation via the Poisson Product model.

## 1.2. Contribution of the Paper

In this paper, we reinforce the foundations of multinomial distribution estimation with a precise characterization of the *dependencies* between  $N_x = \sum_{i=1}^n \mathbb{1}(X_i = x)$  across different  $x \in \mathcal{X}$  (rather than assuming independence and using the Poisson approximation). The theoretical analysis is based on the *expected value* of the frequency of frequencies  $\mathbb{E}[\Phi_k] = f_k(n)$  between different  $k$  and  $n$ , which is

$$\frac{f_k(n)}{\binom{n}{k}} = \frac{f_k(n+1)}{\binom{n+1}{k}} - \frac{f_{k+1}(n+1)}{\binom{n+1}{k+1}}. \quad (2)$$

Exploring this new theoretical tool, we bring two contributions to the estimation of the total probability mass  $M_k$  for any  $k : 0 \leq k \leq n$ . Firstly, we show *exactly to what extent*  $\mathbb{E}[M_k]$  can be estimated from the sample  $X^n$  and *how much remains* to be estimated from the underlying distribution  $p$  and the number of elements  $|\mathcal{X}|$ . Specifically, we show the following.

### Theorem 1.1.

$$\mathbb{E}[M_k] = \binom{n}{k} \left[ \sum_{i=1}^{n-k} (-1)^{i-1} f_{k+i}(n) \right] / \binom{n}{k+i} + R_{n,k}$$

where  $R_{n,k} = \binom{n}{k} (-1)^{n-k} f_{n+1}(n+1)$  is the remainder.

This decomposition shows that the GT estimator is the *first term* of  $\mathbb{E}[M_k]$  using the plug-in estimator  $\Phi_1$  for  $f_1(n)$ . Hence, it gives the *exact bias* of the GT estimator in the multinomial setting (which would incorrectly be identified as *unbiased* using the Poisson approximation).

Our second contribution is the development of a class of natural estimators. We start by defining a nearly unbiased estimator  $\hat{M}_k^B = \binom{n}{k} \left[ \sum_{i=1}^{n-k} (-1)^{i-1} \Phi_{k+i} / \binom{n}{k+i} \right]$  that uses the plug-in estimator  $\Phi_i$  for  $f_i(n)$  in Theorem 1.1 and drops  $R_{n,k}$ . While the bias of  $\hat{M}_k^B$  decays exponentially, the variance of  $\hat{M}_k^B$  is too high to be practical.

We observe that we can cast the estimation of the expected total mass as an optimization problem. From Theorem 1.1 and Eqn. 2, we can see that there are many *representations* of  $\mathbb{E}[M_k]$ , all of which suggest different estimators for  $\mathbb{E}[M_k]$ . We introduce a deterministic method to construct a unique estimator from a representation, and show how to estimate the mean squared error (MSE) for such an estimator. Equipped with a large *search space* of representations and a *fitness function* to estimate the MSE of a candidate estimator, we develop a genetic algorithm that finds an estimator with a minimal MSE (and at most N terms).

We compare the performance of the minimal-bias estimator  $\hat{M}_k^B$  and the minimal-MSE estimators discovered by our genetic algorithm to the that of the widely used GT estimator on a variety of multinomial distributions used for evaluation in previous work. Our results show that 1) the minimal-bias estimator has a substantially smaller bias than the GT estimator by thousands of order of magnitude, 2) Our genetic algorithm can produce estimators with MSE smaller than the GT estimator over 96% of the time when there are at least as many samples as classes; their MSE is roughly 80% of the GT estimator. We also publish all data and scripts to reproduce our results.

## 2. Dependencies Between Frequencies $N_x$

Firstly, we propose a new, distribution-free<sup>2</sup> methodology for reasoning about properties of estimators of the missing and total probability masses for multinomial distributions. The *main challenge* for the statistical analysis of  $M_k$  has been reasoning in the presence of dependencies between frequencies  $N_x$  for different elements  $x \in \mathcal{X}$ . As discussed in Section 1.1, a Poisson approximation with parameter  $\lambda_x = p_x n$  is often used to render these frequencies as independent (Orlitsky & Suresh, 2015; Orlitsky et al., 2016; Acharya et al., 2013; Efron & Thisted, 1976; Valiant & Valiant, 2016; Good, 1953; Good & Toulmin, 1956; Hao

<sup>2</sup>A *distribution-free analysis* is free of assumptions about the shape of the probability distribution generating the sample. In this case, we make no assumptions about parameters  $p$  or  $n$ .

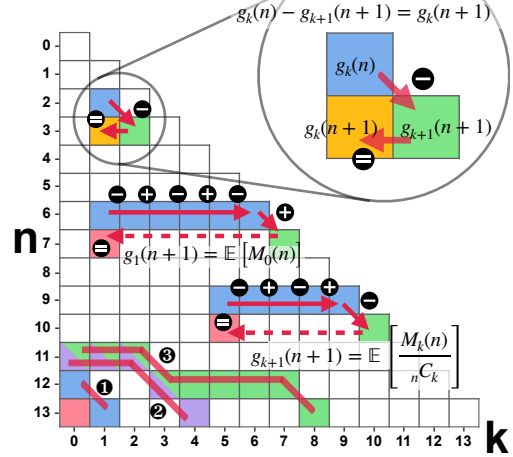


Figure 1. Lower triangle matrix of  $g_k(n)$ .

& Li, 2020). In the following, we tackle this challenge by formalizing these dependencies between frequencies. Thus, we establish a link between the expected values of the corresponding total probability masses.

### 2.1. Dependency Among Frequencies

Recall that the expected value  $f_k(n)$  of the number of elements  $\Phi_k$  that appear exactly  $k$  times in the sample  $X^n$  is defined as  $f_k(n) = \sum_{x \in \mathcal{X}} \binom{n}{k} p_x^k (1 - p_x)^{n-k}$ . For convenience, let  $g_k(n) = f_k(n) / \binom{n}{k}$ . We notice the following relationship among  $k$  and  $n$ :

$$\begin{aligned} g_k(n+1) &= \sum_{x=1}^S p_x^k (1 - p_x)^{n-k} \cdot (1 - p_x) \\ &= g_k(n) - g_{k+1}(n+1) \\ &= \sum_{i=0}^{n-k} (-1)^i g_{k+i}(n) + (-1)^{n-k+1} g_{n+1}(n+1) \end{aligned} \quad (3)$$

We can now write the expected value  $\mathbb{E}[M_k]$  of the total probability mass in terms of the frequencies with which different elements  $x \in \mathcal{X}$  have been observed in the sample  $X^n$  of size  $n$  as follows

$$\begin{aligned} \mathbb{E}[M_k] &= \sum_{x \in \mathcal{X}} \binom{n}{k} p_x^{k+1} (1 - p_x)^{n-k} \\ &= \binom{n}{k} g_{k+1}(n+1) \end{aligned} \quad (4)$$

$$= \binom{n}{k} \left[ \sum_{i=1}^{n-k} (-1)^{i-1} g_{k+i}(n) \right] + R_{n,k} \quad (5)$$

where  $R_{n,k} = \binom{n}{k} (-1)^{n-k} f_{n+1}(n+1)$  is a remainder term. **This demonstrates Theorem 1.1.**

Figure 1 illustrates the relationship between the expected frequency of frequencies  $f_k(n) = g_k(n) / \binom{n}{k}$ , the frequency

$k$ , and the sample size  $n$ . The  $y$ - and  $x$ -axis represents the sample size  $n$  and the frequency  $k$ , respectively. As per Eqn. (3), for every 2x2 lower triangle matrix, the value of the lower left cell ( $g_k(n+1)$ ) is value of the upper left cell ( $g_k(n)$ ) minus the value of the lower right cell ( $g_{k+1}(n+1)$ ).

We can use this visualization to quickly see how to rewrite  $g_k(n)$  as an alternating sum of values of the cells in the upper row, starting from the cell in the same column to the rightmost cell, and adding/subtracting the value of the rightmost cell in the current row. For instance, the value  $g_0(13)$  in the bottom red cell (row  $n = 13$ , column  $k = 0$ ) in the figure can be equally calculated as the various linear combinations of its surrounding cells: (1) with  $g_0(12)$  and  $g_1(13)$  (blue colored), (2) with  $g_0(11)$ ,  $g_1(11)$ ,  $\dots$ ,  $g_4(13)$  (purple colored), or (3) with  $g_0(11)$ ,  $g_1(11)$ ,  $\dots$ ,  $g_8(13)$  (green colored).

**Missing Mass.** The missing probability mass  $M_0$  gives the proportion of *all possible observations* for which the elements  $x \in \mathcal{X}$  have *not* been observed in  $X^n$ . By Eqn. (4) and (5), the expected value of  $M_0$  is

$$\begin{aligned} \mathbb{E}[M_0] &= g_1(n+1) \\ &= \left[ \sum_{k=1}^n (-1)^{k-1} g_k(n) \right] + (-1)^n f_{n+1}(n+1). \end{aligned}$$

The values in the second column of Figure 1 ( $k = 1$ ) represents the expected values of missing mass;  $\mathbb{E}[M_0]$  being the cumulative sum of  $(-1)^{k-1} g_k(n)$  is intuitively clear from the figure (the red cell in the row  $n = 7$ ). It is here where we observe that  $\mathbb{E}[M_0] = g_1(n+1)$  is *almost entirely determined* by the  $g_*(n)$ , the expected frequencies of frequencies in the sample  $X^n$ , and *not* by the number of elements  $|\mathcal{X}|$  or their underlying distribution  $p$ . In fact, the influence of  $p$  in the remainder term decays exponentially, i.e.,  $f_{n+1}(n+1) = \sum_{x \in \mathcal{X}} p_x^{n+1} \leq \sum_{x \in \mathcal{X}} (e^{1-p_x})^{-n-1}$  which is dominated by the discovery probability of the most abundant element  $\max(p)$ .

**Total Mass.** Similarly, the expected value of the total probability mass  $\mathbb{E}[M_k]$  (the red cell in the row  $n = 10$ ), which is equal to  $\binom{n}{k} g_{k+1}(n+1)$ , is almost entirely determined by the expected frequencies of the sample  $X^n$  with remainder  $R_{n,k} = \binom{n}{k} \sum_{x \in \mathcal{X}} p_x^{n+1}$ .

### 3. A Large Class of Estimators

From the representation of  $\mathbb{E}[M_k]$  in terms of frequencies in Eqn. (4) and the relationship across frequencies in Eqn. (3), we can see that there is a large number of representations of the expected total probability mass  $\mathbb{E}[M_k]$ . Each representation might suggest different estimators.

We start by defining the minimal bias estimator  $\hat{M}_k^B$  from the representation in Eqn. (5) and explore its properties.

#### 3.1. Estimator with Exponentially Decaying Bias

Let

$$\hat{M}_k^B = - \binom{n}{k} \sum_{i=1}^{n-k} \frac{(-1)^i \Phi_{k+i}}{\binom{n}{k+i}}$$

**Bias.** For some constant  $k : 0 \leq k \leq n$  and some constant  $c > 1$ , the bias of  $\hat{M}_k^B$  is in the order of  $\mathcal{O}(n^k c^{-n})$ , i.e.,

$$\begin{aligned} \mathbb{E}[\hat{M}_k^B - M_k] &= -R_{n,k} = (-1)^{n-k-1} \binom{n}{k} \sum_{x \in \mathcal{X}} p_x^{n+1} \\ \left| \mathbb{E}[\hat{M}_k^B - M_k] \right| &\leq \binom{n}{k} \sum_{x \in \mathcal{X}} c_x^{-n} \leq n^k \sum_{x \in \mathcal{X}} c_x^{-n} \end{aligned}$$

where  $c_x > 1$  for all  $x \in \mathcal{X}$  are constants.

**Variance.** The variance of  $\hat{M}_k^B$  is given by the variances and covariances of the frequencies  $\Phi_{k+i}$  for  $i = 1, \dots, n-k$ . Under the certain conditions, the variance of  $\hat{M}_k^B$  also decays exponentially in  $n$ .

**Theorem 3.1.** *The variance of  $\hat{M}_k^B$  decreases exponentially with  $n$  if  $p_{\max} < 0.5$  or  $\frac{(1-p_{\max})(1-p_{\min})}{p_{\max}} < 1$ , where  $p_{\max} = \max_{x \in \mathcal{X}} p_x$  and  $p_{\min} = \min_{x \in \mathcal{X}} p_x$ .*

The proof is postponed to Appendix B.

**Comparison to Good-Turing (GT).** The bias of  $\hat{M}_k^B$  not only decays exponentially in  $n$  but is also *smaller* than that of GT estimator  $\hat{M}_k^G$  by an exponential factor. For a simpler variant of GT estimator,  $\hat{M}_k^{G'} = \frac{k+1}{n-k} \Phi_{k+1}$  (suggested in (McAllester & Schapire, 2000)), which corresponds to the first term in the expected total probability mass  $\mathbb{E}[M_k]$  in Eqn. (5), we show that its bias is *larger by an exponential factor* than the absolute bias of  $\hat{M}_k^B$ . To see this, we provide bounds on the individual sums and then on the bias ratio:

$$\text{Bias}_{G'} = \mathbb{E}[\hat{M}_k^{G'} - M_k] \geq \binom{n}{k} p_{\min}^{k+2} (1-p_{\min})^{n-k-1} \quad (6)$$

$$|\text{Bias}_B| = \left| \mathbb{E}[\hat{M}_k^B - M_k] \right| \leq \binom{n}{k} S p_{\max}^{n+1} \quad (7)$$

where  $S = |\mathcal{X}|$ , such that

$$\left| \frac{\text{Bias}_{G'}}{\text{Bias}_B} \right| \geq \frac{p_{\min}^{k+2}}{S p_{\max}^{k+2}} \left( \frac{1-p_{\min}}{p_{\max}} \right)^{n-k-1}.$$

Notice that  $(1-p_{\min})/p_{\max} > 1$  for all distributions over  $\mathcal{X}$ , except where  $S = 2$  and  $p = \{0.5, 0.5\}$ . The same can be shown for the original GT estimator  $\hat{M}_k^G = \frac{k+1}{n} \Phi_{k+1}$  for a sufficiently large sample size (see Appendix A).

**Example** (Missing mass for the uniform). Suppose, we seek to estimate the missing mass from a sequence of elements  $X^n$  observed in  $n$  samples from the uniform distribution;  $p_x = 1/S$  for all  $x \in \mathcal{X}$ .  $\hat{M}_0^G$  overestimates  $M_0$  on the

average by  $(S-1)^{n-1}/S^n$  while our new estimator  $\hat{M}_0^B$  has a bias of  $(-1)^n/S^n$ . Hence, for the uniform distribution, our estimator exhibits a bias that is lower by a factor of  $1/(S-1)^{n-1}$ .

While the bias of our estimator  $\hat{M}_k^B$  is lower than that of  $\hat{M}_0^G$  by an exponential factor, the variance is higher. The variance of  $\hat{M}_k^B$  depends on the variances of and covariances between  $\Phi_{k+1}, \Phi_{k+2}, \dots, \Phi_n$ , i.e.,

$$\text{Var}(\hat{M}_k^B) = \sum_{i=1}^{n-k} c_i^2 \text{Var}(\Phi_{k+i}) + \sum_{i \neq j} (-1)^{i+j} c_i c_j \text{Cov}(\Phi_{k+i}, \Phi_{k+j}), \quad (8)$$

where  $c_i = \binom{n}{k} / \binom{n}{k+i}$ . In contrast, the variance of  $\hat{M}_k^G$  depends only on the variance of  $\Phi_{k+1}$ . In the empirical study, we investigate the difference of the two estimators in terms of the bias and the variance.

### 3.2. Estimation with Minimal MSE as Search Problem

There are many representations of  $\mathbb{E}[M_k] = \binom{n}{k} g_{k+1}(n+1)$  that can be constructed by recursively rewriting terms according to the dependency among frequencies we identified (cf. Eqn. (3 & 4)). The representation used to construct our minimal-bias estimator  $\hat{M}_k^B$  was one of them. However, we notice that the variance of  $\hat{M}_k^B$  is too high to be practical. To find a representation from which an estimator with a minimal mean squared error (MSE) can be derived, we cast the efficient estimation of  $M_k$  as an *optimization problem*. To efficiently navigate the large search space of representations of  $\mathbb{E}[M_k]$ , we develop a genetic algorithm.

**Search space.** Let  $\mathbb{E}[M_k]$  be *represented* by a suitable choice of coefficients  $\alpha_{i,j}$  such that

$$\mathbb{E}[M_k] = \sum_{i=1}^{n+1} \sum_{j=i}^{n+1} \alpha_{i,j} g_i(j). \quad (9)$$

One representation of  $\mathbb{E}[M_k] = \binom{n}{k} g_{k+1}(n+1)$  is

$$r_0 = \left\{ \alpha_{i,j} = \begin{cases} \binom{n}{k} & \text{for } i = k+1 \text{ and } j = n+1 \\ 0 & \text{otherwise.} \end{cases} \right\} \quad (10)$$

**Mutation.** Given any representation  $r$  of  $\mathbb{E}[M_k]$ , we can construct a new representation  $r'$  of  $\mathbb{E}[M_k]$ , s.t. Eqn. (9) holds by *recursively* considering the following identities:

$$\alpha_{i,j} \cdot g_i(j) = \alpha_{i,j} \cdot ((1-\delta)g_i(j) + \delta g_i(j)) \quad (11)$$

$$= \alpha_{i,j} \cdot (g_i(j+1) + g_{i+1}(j+1)) \quad (12)$$

$$= \alpha_{i,j} \cdot (g_i(j-1) - g_{i+1}(j)) \quad (13)$$

$$= \alpha_{i,j} \cdot (g_{i-1}(j-1) + g_{i-1}(j)) \quad (14)$$

for any choice of  $\delta : 0 \leq \delta \leq 1$ . *Importantly*, after applying these identities, we must work out the new coefficients

### Algorithm 1 Genetic Algorithm

---

**Input:** Target frequency  $k$ , Sample  $X^n$   
**Input:** Iteration limit  $G$ , mutant size  $m$   
 1: Population  $P_0 = \{r_0\}$   
 2: Fitness  $f^{\text{best}} = f_0 = \text{fitness}(r_0)$   
 3: Limit  $G_L = G$   
 4: **for**  $g$  from 1 to  $G_L$  **do**  
 5:    $P = \text{selectTopM}(P_{g-1}, m)$   
 6:    $P' = \text{lapply}(P, \text{mutate})$   
 7:    $P_g = P' \cup \{r_0\} \cup \text{selectTopM}(P_{g-1}, 3)$   
 8:    $f_g = \min(\text{lapply}(P_g, \text{fitness}))$   
 9:   **if**  $(g = G_L) \wedge ((f_g = f_0) \vee (f^{\text{best}} > 0.95 \cdot f_g))$  **then**  
 10:      $G_L = G_L + G$   
 11:      $f^{\text{best}} = f_g$   
 12:   **end if**  
 13: **end for**  
 14: Estimator  $\hat{M}_k^{\text{Evo}} = \text{instantiate}(\text{selectTopM}(P_{G_L}, 1))$   
**Output:** Minimal-MSE Estimator  $\hat{M}_k^{\text{Evo}}$

---

accordingly. For instance, applying Eqn. (11) with  $\delta = 0.5$  and Eqn. (13) to  $r_0$ , we obtain the following representation  $r_1$  of  $\mathbb{E}[M_k]$ :

$$r_1 = \left\{ \alpha_{i,j} = \begin{cases} \binom{n}{k}/2 & \text{for } i = k+1 \text{ and } j = n+1 \\ \binom{n}{k}/2 & \text{for } i = k+1 \text{ and } j = n \\ -\binom{n}{k}/2 & \text{for } i = k+2 \text{ and } j = n+1 \\ 0 & \text{otherwise.} \end{cases} \right\}$$

**Estimator instantiation.** To construct a unique estimator  $\hat{M}_k^r$  of  $M_k$  from a representation  $r$  of  $\mathbb{E}[M_k]$ , we propose a deterministic method. But first, we define our random variables on subsamples of  $X^n$ . For any  $m \leq n$ , let  $N_x(m)$  be the number of times element  $x \in \mathcal{X}$  is observed in the subsample  $X^m = \langle X_1, \dots, X_m \rangle$  of  $X^n$ . Let  $\Phi_k(m)$  be the number of elements appearing exactly  $k$  times in  $X^m$ , i.e.,

$$N_x(m) = \sum_{i=1}^m \mathbb{1}(X_i = x) \quad \text{Note that } N_x = N_x(n).$$

$$\Phi_k(m) = \sum_{x \in \mathcal{X}} \mathbb{1}(N_x(m) = k) \quad \text{Note that } \Phi_k = \Phi_k(n).$$

Hence, given a representation  $r$ , we can construct  $\hat{M}_k^r$  as

$$\hat{M}_k^r = \left[ \sum_{i=1}^n \sum_{j=i}^n \frac{\alpha_{i,j}}{\binom{j}{i}} \Phi_i(j) \right] + \left[ \sum_{i=1}^n \frac{\alpha_{i,n+1}}{\binom{n+1}{i}} \Phi_i \right]$$

Notice that  $\Phi_i(j) / \binom{j}{i}$  is just the plug-in estimator for  $g_i(j)$ .

**Fitness function.** To define the quantity to optimize, any meta-heuristic search requires a fitness function. Our *fitness function* takes a candidate representation  $r$  and returns an estimate of the MSE of the corresponding estimator  $\hat{M}_k^r$ . We decompose the MSE as the sum of its variance and squared bias. For convenience, let  $g_{n+1}(n) = 0$ .

$$\begin{aligned} \text{MSE}(\hat{M}_k^r) &= \left[ \sum_{i=1}^{n+1} \alpha_{i,n+1} [g_i(n+1) - g_i(n)] \right]^2 \\ &+ \sum_{i=1}^n \sum_{j=i}^n \left( \frac{\alpha_{i,j}}{\binom{j}{i}} \right)^2 \text{Var}(\Phi_i(j)) \\ &+ \sum_{i=1}^n \sum_{j=i}^n \sum_{\substack{l=1 \\ l \neq i}}^n \sum_{\substack{m=l \\ m \neq j}}^n \frac{\alpha_{i,j}}{\binom{j}{i}} \frac{\alpha_{l,m}}{\binom{m}{l}} \text{Cov}(\Phi_i(j), \Phi_l(m)) \end{aligned} \quad (15)$$

We expand on the computation of the MSE in Appendix C.

Since the underlying distribution  $\{p_x\}_{x \in \mathcal{X}}$  is *unknown*, we can only *estimate* the MSE. For any element  $x$  that has been observed exactly  $k > 0$  time in the sample  $X^n$ , we use  $\hat{p}_x = \hat{M}_k^G / \Phi_k$  as natural estimator of  $p_x$ , where  $\hat{M}_k^G$  is the GT estimator. To handle unobserved elements ( $k = 0$ ), we first estimate the number of unseen elements  $\mathbb{E}[\Phi_0] = f_0(n)$  using Chao’s nonparametric species richness estimator  $\hat{f}_0 = \frac{n-1}{n} \frac{\Phi_1^2}{2\Phi_2}$  (Chao, 1984), and then estimate the probability of each such unseen element as  $\hat{p}_y = \hat{M}_0^G / \hat{f}_0$ , where  $\hat{M}_0^G$  is the GT estimator. Finally, we plug these estimates into Eqn. (15) to estimate the MSE. It is interesting to note that it is precisely the GT estimator whose MSE our approach is supposed to improve upon.

**Genetic algorithm.** With the required concepts in place, we are ready to introduce our *genetic algorithm* (GA) (Mitchell, 1998). Algorithm 1 sketches the general procedure. Given a target frequency  $k$  (incl.  $k = 0$ ), the sample  $X^n$ , an iteration limit  $G$ , and the number  $m$  of candidate representations to be mutated in every iteration, the algorithm produces an estimator  $\hat{M}_k^{\text{Evo}}$  with minimal MSE.

Starting from the *initial representation*  $r_0$  (Eqn. (10); Line 1), our GA iteratively improves a population of candidate representations  $P_g$ , called *individuals*. For every generation  $g$  (Line 4), our GA selects the  $m$  fittest individuals from the previous generation  $P_{g-1}$  (Line 5), mutates them (Line 6), and creates the current generation  $P_g$  by adding the initial representation  $r_0$  and the Top-3 individuals from the previous generation (Line 7). The initial and previous Top-3 individuals are added to mitigate the risk of convergence to a local optimum. To *mutate* a representation  $r$ , our GA (i) chooses a random term  $r$ , (ii) applies Eqn. (11) where  $\delta$  is chosen uniformly at random, (iii) applies a random identity from Eqn. (12–14), and (iv) adjusts the coefficients for the resulting representation  $r'$  accordingly. The iteration limit  $G_L$  is increased if the current individuals do *not* improve on the initial individual  $r_0$  or *substantially* improve on those discovered recently (Line 9–12).

**Distribution-free.** While our approach itself is *distribution-free*, the output is *distribution-specific*, i.e., the discovered estimator has a minimal MSE on the specific, unknown distribution. We explore this property in our experiments.

## 4. Experiment

We design experiments to evaluate the performance (i) of our minimal-bias estimator  $\hat{M}_k^B$  and (ii) of our the minimal-MSE estimator  $\hat{M}_k^{\text{Evo}}$  that is discovered by our genetic algorithm against the performance of the widely-used Good-Turing estimator  $\hat{M}_k^G$  (Good, 1953).

**Distributions.** We use the same six multinomial distributions that are used in previous evaluations (Orlitsky & Suresh, 2015; Orlitsky et al., 2016; Hao & Li, 2020): a uniform distribution (uniform), a half-and-half distribution where half of the elements have three times of the probability of the other half (half&half), two Zipf distributions with parameters  $s = 1$  and  $s = 0.5$  (zipf-1, zipf-0.5, respectively), and distributions generated by Dirichlet-1 prior and Dirichlet-0.5 prior (diri-1, diri-0.5, respectively).

**Open Science and Replication.** For scrutiny and replicability, we publish all our evaluation scripts at:

<https://anonymous.4open.science/r/Better-Turing-157F>.

### 4.1. Evaluating our Minimal-Bias Estimator

- **RQ1.** How does our estimator for the missing mass  $\hat{M}_0^B$  compare to the Good-Turing estimator  $\hat{M}_0^G$  in terms of bias as a function of sample size  $n$ ?
- **RQ2.** How does our estimator for the total mass  $\hat{M}_k^B$  compare to the Good-Turing estimator  $\hat{M}_k^G$  in terms of bias as a function of frequency  $k$ ?
- **RQ3.** How do the estimators compare in terms of variance and mean-squared error?

We focus specifically on the *bias* of  $\hat{M}_k^B$ , i.e., the average difference between the estimate and the expected value  $\mathbb{E}[M_k]$ . We expect that the bias of the *missing mass* estimator  $\hat{M}_0^B$  as a function of  $n$  across different distributions provides empirical insight for our claim that how much is unseen chiefly depends on information about the seen.

**RQ1.** Figure 2a illustrates how fast our estimator  $\hat{M}_k^B$  and the baseline estimator  $\hat{M}_k^G$  (GT) approach the expected missing mass  $\mathbb{E}[M_0]$  as a function of sample size  $n$ . As it might difficult for the reader to discern differences across distributions for the baseline estimator, we refer to Figure 2b, where we zoom into a relevant region.

The *magnitude* of our estimator’s bias is significantly smaller than the magnitude of GT’s bias *for all distributions* (by thousands of orders of magnitude).<sup>3</sup> Figure 2a also nicely illustrates the *exponential decay* of our estimator in terms of  $n$  and how our estimator is less biased than GT by an exponential factor. In Figure 2b, we can observe that GT’s bias also decays exponentially, although not nearly at the rate of our estimator.

<sup>3</sup>Recall that the plot shows the *logarithm* of the absolute bias.

## How Much is Unseen Depends Chiefly on Information About the Seen

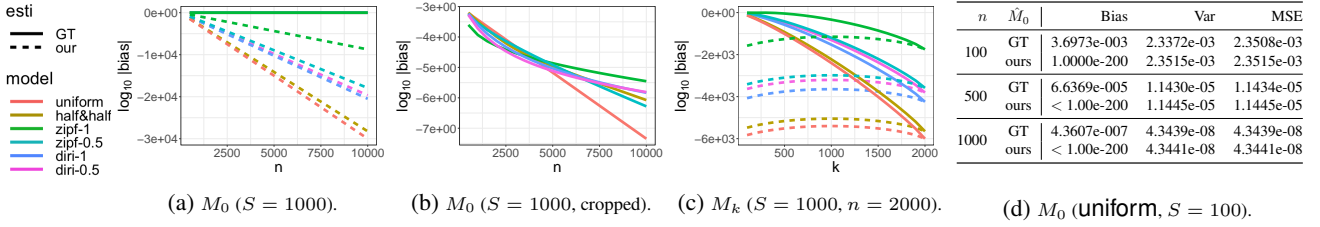


Figure 2. Results for our minimal-bias estimator  $\hat{M}_k^B$ . For our and baseline estimators, we show the logarithm of the absolute bias (a&b) as a function of  $n$  for  $k=0$  and (c) as a function of  $k$  for  $n=2000$ . We also show (d) the bias, variance, and MSE of our and baseline estimator for three values of  $n$ . More plots can be found in Appendix E.

Table 1. The MSE of the best evolved estimator  $M_0^{Evo}$  and GT estimator  $\hat{M}_0^G$  for the missing mass  $M_0$ , the success rate  $\hat{A}_{12}$ , and the ratio (Ratio,  $MSE(\hat{M}_0^{Evo})/MSE(M_0^G)$ ) for three sample sizes  $n$  and six distributions with support size  $S=200$ .

Dist.	$n = S/2$				$n = S$				$n = 2S$			
	$MSE(\hat{M}_0^G)$	$MSE(M_0^{Evo})$	$\hat{A}_{12}$	Ratio	$MSE(\hat{M}_0^G)$	$MSE(M_0^{Evo})$	$\hat{A}_{12}$	Ratio	$MSE(\hat{M}_0^G)$	$MSE(M_0^{Evo})$	$\hat{A}_{12}$	Ratio
uniform	3.32e-03	2.04e-03	0.95	61%	1.17e-03	8.90e-04	0.99	76%	2.01e-04	1.70e-04	0.93	84%
half&half	3.33e-03	1.97e-03	0.96	59%	1.09e-03	8.58e-04	0.99	78%	2.11e-04	1.72e-04	1.00	81%
zipf-1	2.32e-03	2.41e-03	0.74	103%	8.16e-04	7.24e-04	0.88	88%	2.39e-04	2.11e-04	0.96	88%
zipf-0.5	3.23e-03	2.29e-03	0.89	71%	1.09e-03	8.52e-04	0.97	78%	2.30e-04	1.93e-04	1.00	83%
diri-1	2.99e-03	2.36e-03	0.85	78%	8.88e-04	6.65e-04	1.00	74%	1.96e-04	1.65e-04	0.96	84%
diri-0.5	2.55e-03	1.81e-03	0.94	71%	6.88e-04	4.86e-04	0.98	70%	1.61e-04	1.31e-04	0.93	81%
Avg.			0.88	74%			0.96	77%			0.96	84%

In terms of distributions, a closer look at the performance differences confirms our suspicion that the bias of our estimator is strongly influenced by the probability  $p_{\max}$  of the most abundant element while the bias of GT is strongly influenced by the probability  $p_{\min}$  of the rarest element. In fact, by Eqn. (7) the absolute bias of our estimator is minimized when  $p_{\max}$  is minimized. By Eqn. (6), GT’s bias is minimized if  $p_{\min}$  is maximized. Since both is true for the uniform, both estimators exhibit the lowest bias for the uniform across all six distributions. GT performs similar on all distributions apart from the uniform (where bias seems minimal) and zipf-1 (where bias is maximized). For our estimator, if we ranked the distributions by values of  $p_{\max}$  with the smallest value first (uniform, half&half, zipf-0.5, zipf-1),<sup>4</sup> we would arrive at the same ordering in terms of performance of our estimator as shown in Figure 2a.

**RQ2.** Figure 2c illustrates for both estimators of the total mass  $M_k$  how the bias behaves as  $k$  varies between 0 and  $n=2000$  when  $S=1000$ . The trend is clear; the bias of our estimator is strictly smaller than the bias of GT for all  $k$  and all the distributions. The difference is the most significant for rare elements (small  $k$ ) and gets smaller as  $k$  increases. The bias of our estimator is maximized when  $k=1000=0.5n$ , the bias for GT when  $k=0$ .

**RQ3.** Table 2d shows variance and MSE of both estimators for the missing mass  $M_0$  for the uniform and three values of  $n$ . As we can see, the MSE of our estimator is approxi-

mately the same as that of GT. The reason is that the MSE is dominated by the variance. We make the same observation for all other distributions (see Appendix E). The MSEs of both estimators are comparable.

### 4.2. Evaluating our Estimator Discovery Algorithm

- **RQ1** (Effectiveness). *How does our estimator for the missing mass  $\hat{M}_0^{Evo}$  compare to the Good-Turing estimator  $\hat{M}_0^G$  in terms of MSE?*
- **RQ2** (Efficiency). *How long does it take for our genetic algorithm to generate an estimator  $M_k^{Evo}$  given a sample?*
- **RQ3** (Distribution-awareness). *How well does an estimator discovered from a sample from one distribution perform on another distribution in terms of MSE?*

To handle the randomness in our evaluation, we repeat each experiment 100 times: 10 runs of the GA with 10 different samples  $X^n$ .<sup>5</sup> More details about our experimental setup can be found in Appendix D.

**RQ.1** (Effectiveness). Table 1 shows average MSE of the estimator  $M_0^{Evo}$  discovered by our genetic algorithm and that of the GT estimator  $\hat{M}_0^G$  for the missing mass  $M_0$  across three sample sizes. We measure effect size using Vargha-Delaney  $\hat{A}_{12}$  (Vargha & Delaney, 2000) (success rate), i.e., the probability that the MSE of the estimator discovered by our genetic algorithm has a smaller MSE than the GT

<sup>4</sup>diri-1 and diri-0.5 are not considered in the order because multiple distributions are sampled from the Dirichlet prior.

<sup>5</sup>For diri-1, diri-0.5, each of the ten samples  $X^n$  is sampled from 10 distributions sampled from the Dirichlet prior with the same parameter  $\alpha=0.5, 1$ , respectively.

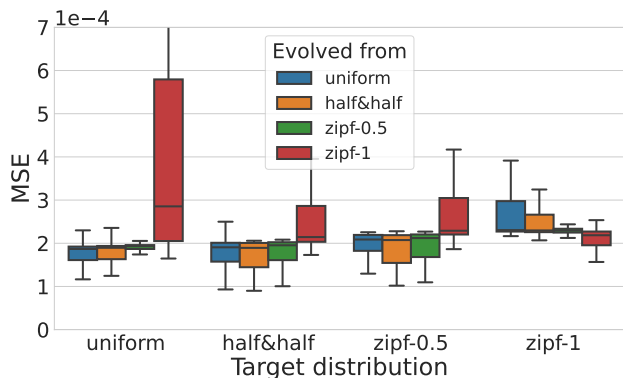


Figure 3. The MSE of an estimator discovered using a sample from one distribution (individual boxes) applied to another target distribution (clusters of boxes).

estimator (*larger is better*). Moreover, we measure the MSE of our estimator as a proportion of the MSE of GT, called *ratio* (*smaller is better*). Results for other support sizes  $S$  can be found in Appendix E.

Overall, the estimator discovered by our GA performs *significantly* better than GT estimator in terms of MSE (avg.  $\hat{A}_{12} > 0.9$ ; *ratio*  $< 85\%$ ). The performance difference increases with sample size  $n$ . When the sample size is twice the support size ( $n = 2S$ ), in 96% of runs our discovered estimator performs better. The average MSE of our estimator is somewhere between 70% and 88% of the MSE of GT. The high success rate and the low ratio of the MSE shows that the GA is effective in finding the estimator with the minimal MSE for the missing mass  $M_0$ . A Wilcoxon signed-rank test shows that all performance differences are statistically significant at  $\alpha < 10^{-9}$ .

In terms of distributions, the performance of our estimator is similar across all distributions, showing the generality of our algorithm. The only exception is the zipf-1, where the success rate is lower than for other distribution for  $n = S/2$  and  $S$ , and the average ratio is 103% (yet, the median ratio is 85%) for  $n = S/2$ . The potential reason for this is due to the overfitting to the approximated distribution  $\hat{p}_x$ . Since the zipf-1 is the most skewed distribution, there are more elements unseen in the sample than in other distributions, which makes the approximated distribution  $\hat{p}_x$  less accurate. Yet, the performance already improves and become similar to other distributions for  $n = S$  and  $n = 2S$ .

**RQ.2** (Efficiency). The time it takes to discover the estimator is reasonable. To compute an estimator for Table 1, it takes about seven (7) minutes on average and five (5) minutes on median. The average time for each iteration is 1.25s (median: 0.92).

**RQ.3** (Distribution-awareness). Figure 3 shows the performance of an estimator discovered from a sample from one distribution (source) when applied to another distribution

(target). Applying an estimate from the zipf-1 on the zipf-1 gives the optimal MSE (right-most red box). However, applying an estimator from the zipf-1 on the uniform (left red box) yields a huge increase in variance. In terms of effect size, we measure a Vargha Delaney  $\hat{A}_{12} > 0.84$  between the “home” and “away” estimator. While each of the uniform and half&half also shows that the home estimator performs best on the home distribution ( $\hat{A}_{12} = 0.68$  (medium), 0.58 (small), respectively), the difference between the estimators from uniform, half&half, and zipf-0.5 is less significant. Perhaps unsurprisingly, an estimator performs optimal when the source of the samples is similar to the target distribution.

**Summary.** To summarize, our GA is effective in finding the estimator with the minimal MSE for the missing mass  $M_0$  with the smaller MSE than GT estimator  $\hat{M}_0^G$  for all distributions and sample sizes. The effect is substantial and significant and the average decrease of the MSE is roughly one fifth against GT estimator  $\hat{M}_0^G$ .

## 5. Discussion

**Beyond the General Estimator.** In this study, we propose a meta-level estimation methodology that can be applied to a set of samples from a specific unknown distribution. The conventional approach is to develop *an estimator* for an arbitrary distribution. Yet, each distributions has its own characteristics, and, because of that, the manner of the (frequencies of) frequencies of the classes in the sample can be differ from, for example, the uniform distribution to the Zipf distribution. In contrast to the conventional approach, we propose a *distribution-free* methodology to discover the a *distribution-specific* estimator with low MSE (given only the sample). Note that, while we use the genetic algorithm to discover the estimator, any optimization method can be used to discover the estimator, for instance, a constrained optimization solver.

**Extrapolating the Future Sampling.** Estimating the number of unseen species is a well-known problem in many scientific fields, such as ecology, linguistics, and machine learning. Given  $n$  samples, the expected number of hitherto unseen species that would be uncovered if  $t$  times more samples were taken is  $\mathbb{E}[U(t)] = f_0(n) - f_0(n + nt)$ . Good & Toulmin (1956) proposed a seminal estimator using the frequencies of frequencies  $\Phi_k$ , similar to the Good-Turing estimator. Until recently, various subsequent studies have been conducted to improve the estimator (Efron & Thisted, 1976; Orlitsky et al., 2016; Hao & Li, 2020), while most of them still relies on the Poisson approximation to design the estimator. We believe that our analysis can be extended to the Good-Toulmin estimator seeking more accurate estimators for  $U(t)$ .



## References

- Acharya, J., Jafarpour, A., Orlitsky, A., and Suresh, A. T. Optimal probability estimation with applications to prediction and classification. In Shalev-Shwartz, S. and Steinwart, I. (eds.), *Proceedings of the 26th Annual Conference on Learning Theory*, volume 30 of *Proceedings of Machine Learning Research*, pp. 764–796, Princeton, NJ, USA, 12–14 Jun 2013. PMLR.
- Chao, A. Nonparametric estimation of the number of classes in a population. *Scandinavian Journal of statistics*, pp. 265–270, 1984.
- Chao, A. and Jost, L. Coverage-based rarefaction and extrapolation: standardizing samples by completeness rather than size. *Ecology*, 93 12:2533–47, 2012.
- Druk, E. and Mansour, Y. Concentration bounds for unigrams language model. In Shawe-Taylor, J. and Singer, Y. (eds.), *Learning Theory*, pp. 170–185, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. ISBN 978-3-540-27819-1.
- Efron, B. and Thisted, R. Estimating the number of unseen species: How many words did shakespeare know? *Biometrika*, 63(3):435–447, 1976. ISSN 00063444. URL <http://www.jstor.org/stable/2335721>.
- Good, I. J. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40(3-4): 237–264, 1953.
- Good, I. J. and Toulmin, G. H. The number of new species, and the increase in population coverage, when a sample is increased. *Biometrika*, 43(1/2):45–63, 1956. ISSN 00063444. URL <http://www.jstor.org/stable/2333577>.
- Hao, Y. and Li, P. Optimal prediction of the number of unseen species with multiplicity. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 8553–8564. Curran Associates, Inc., 2020. URL <https://proceedings.neurips.cc/paper/2020/file/618790ae971abb5610b16c826fb72d01-Paper.pdf>.
- Juang, B.-H. and Lo, S. On the bias of the turing-good estimate of probabilities. *IEEE Transactions on signal processing*, 2(42):496–498, 1994.
- McAllester, D. and Schapire, R. E. On the convergence rate of good-Turing estimators. In *Proceedings of the 13th Annual Conference on Computational Learning Theory*, pp. 1–6. Morgan Kaufmann, San Francisco, 2000.
- Mitchell, M. *An introduction to genetic algorithms*. MIT press, 1998.
- Orlitsky, A. and Suresh, A. T. Competitive distribution estimation: Why is good-turing good. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL <https://proceedings.neurips.cc/paper/2015/file/d759175de8ea5b1d9a2660e45554894f-Paper.pdf>.
- Orlitsky, A., Santhanam, N. P., and Zhang, J. Always good turing: Asymptotically optimal probability estimation. *Science*, 302(5644):427–431, 2003. doi: 10.1126/science.1088284.
- Orlitsky, A., Suresh, A. T., and Wu, Y. Optimal prediction of the number of unseen species. *Proceedings of the National Academy of Sciences*, 113(47):13283–13288, 2016. doi: 10.1073/pnas.1607774113. URL <https://www.pnas.org/doi/abs/10.1073/pnas.1607774113>.
- Painsky, A. Convergence guarantees for the good-turing estimator. *Journal of Machine Learning Research*, 23(279):1–37, 2022. URL <http://jmlr.org/papers/v23/21-1528.html>.
- Valiant, G. and Valiant, P. Instance optimal learning of discrete distributions. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, STOC ’16, pp. 142–155, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450341325. doi: 10.1145/2897518.2897641. URL <https://doi.org/10.1145/2897518.2897641>.
- Vargha, A. and Delaney, H. D. A critique and improvement of the cl common language effect size statistics of mcgraw and wong. *Journal of Educational and Behavioral Statistics*, 25(2):101–132, 2000.

## A. Comparing the Bias of the Estimators

In Section 3.1, we have shown that the bias of a simpler variant of GT,  $\hat{M}_k^{G'} = \frac{k+1}{n-k} \Phi_{k+1}$ , is larger by an exponential factor than the absolute bias of our minimal bias estimator  $\hat{M}_k^B$ . In this section, we show that the bias of the original GT estimator  $\hat{M}_k^G = \frac{k+1}{n} \Phi_{k+1}$  is also larger by an exponential factor than the absolute bias of  $\hat{M}_k^B$  for a sufficiently larger sample size. Recall that

$$\text{Bias}_{G'} = \mathbb{E} \left[ \hat{M}_k^G - M_k \right] = \frac{k+1}{n-k} f_{k+1}(n) - \binom{n}{k} g_{k+1}(n+1) = \sum_x \binom{n}{k} p_x^{k+2} (1-p_x)^{n-k-1}, \quad (16)$$

and

$$\text{Bias}_G = \mathbb{E} \left[ \hat{M}_k^G - M_k \right] = \binom{n}{k} g_{k+2}(n+1) - \frac{k(k+1)}{n(n-k)} f_{k+1}(n) \quad (17)$$

$$= \binom{n}{k} g_{k+2}(n+1) - \binom{n-1}{k-1} g_{k+1}(n) \quad (18)$$

$$= \sum_x p_x^{k+2} (1-p_x)^{n-k-1} \left( \binom{n}{k} - \frac{1}{p_x} \cdot \binom{n-1}{k-1} \right) \quad (19)$$

$$= \sum_x \binom{n}{k} p_x^{k+2} (1-p_x)^{n-k-1} \left( 1 - \frac{k}{n \cdot p_x} \right) \quad (20)$$

$$\geq \left( 1 - \frac{k}{n \cdot p_{\max}} \right) \text{Bias}_{G'}, \quad (21)$$

where  $1 - \frac{k}{n \cdot p_{\min}} > 0$  when  $n$  is sufficiently large. Above inequality leads to the following:

$$\frac{\text{Bias}_G}{\text{Bias}_{G'}} \geq \left( 1 - \frac{k}{n \cdot p_{\max}} \right), \quad \text{while} \quad \frac{\text{Bias}_B}{\text{Bias}_{G'}} \leq \frac{S p_{\max}^{k+2}}{p_{\min}^{k+2}} \left( \frac{1-p_{\min}}{p_{\max}} \right)^{-n+k+1}, \quad (22)$$

which proves our claim.

## B. Bounding the Variance and the MSE of $\hat{M}_k^B$

The MSE of an estimator  $\hat{e}$  for an estimand  $e$  is defined as  $MSE(\hat{e}) = \mathbb{E}[(\hat{e} - e)^2] = \text{Var}(\hat{e}) + \text{Bias}^2(\hat{e})$ . As we have shown the bias of  $\hat{M}_k^B$  in Section 3.1, the remaining part to compute the MSE of  $\hat{M}_k^B$  is to compute its variance.

The variance of the linear combination of random variables is given by

$$\text{Var} \left( \sum_i c_i X_i \right) = \sum_i c_i^2 \text{Var}(X_i) + \sum_{i \neq j} c_i c_j \text{Cov}(X_i, X_j). \quad (23)$$

Therefore, the variance and the covariance of  $\Phi_i(n)$ s are the missing pieces to compute the variance of  $\hat{M}_k^B$ .

**Theorem B.1.** *Given the multinomial distribution  $p = (p_1, \dots, p_S)$  with support size  $S$ , the variance of  $\Phi_i = \Phi_i(n)$  from  $n$  samples  $X^n$  is given by*

$$\text{Var}(\Phi_i(n)) = \begin{cases} f_i(n) - f_i(n)^2 + \sum_{x \neq y} \frac{n!}{i!^2 (n-2i)!} p_x^i p_y^i (1-p_x-p_y)^{n-2i} & \text{if } 2i \leq n, \\ f_i(n) - f_i(n)^2 & \text{otherwise.} \end{cases} \quad (24)$$

*Proof.*

$$\text{Var}(\Phi_i) = \mathbb{E}[\Phi_i^2] - \mathbb{E}[\Phi_i]^2 \quad (25)$$

$$\mathbb{E}[\Phi_i^2] = \mathbb{E}\left[\left(\sum_x \mathbb{1}(N_x = i)\right)^2\right] \quad (26)$$

$$= \mathbb{E}\left[\sum_x \mathbb{1}(N_x = i) + \sum_{x \neq y} \mathbb{1}(N_x = i \wedge N_y = i)\right] \quad (27)$$

$$= \begin{cases} f_i(n) + \sum_{x \neq y} \frac{n!}{i!^2(n-2i)!} p_x^i p_y^i (1-p_x-p_y)^{n-2i} & \text{if } 2i \leq n, \\ f_i(n) & \text{otherwise.} \end{cases} \quad (28)$$

$$\therefore \text{Var}(\Phi_i) = \begin{cases} f_i(n) + \sum_{x \neq y} \frac{n!}{i!^2(n-2i)!} p_x^i p_y^i (1-p_x-p_y)^{n-2i} - f_i(n)^2 & \text{if } 2i \leq n, \\ f_i(n) - f_i(n)^2 & \text{otherwise.} \end{cases} \quad (29)$$

□

Now we compute the upper bound of the variance of  $\hat{M}_k^B$ .

**Lemma B.2.**

$$\text{Var}(\Phi_i) \begin{cases} \leq S f_i(n) - f_i(n)^2 & \text{if } 2i \leq n. \\ = f_i(n) - f_i(n)^2 & \text{otherwise.} \end{cases} \quad (30)$$

*Proof.* From Theorem B.1,

$$\mathbb{E}[\Phi_i^2] = f_i(n) + \mathbb{E}\left[\sum_{x \neq y} \mathbb{1}(N_x = i \wedge N_y = i)\right] \quad (31)$$

$$\leq f_i(n) + (S-1) \mathbb{E}\left[\sum_x \mathbb{1}(N_x = i)\right] \quad (32)$$

$$= S f_i(n) \quad (\text{if } 2i \leq n). \quad (33)$$

$$(34)$$

The lemma directly follows from the above inequality. □

**Lemma B.3.**

$$g_i(n) \leq S \cdot \beta_{\min}^{-n} o_{\max}^i,$$

where  $S = |\mathcal{X}|$ ,  $p_{\max} = \max_{x \in \mathcal{X}} p_x$ ,  $\beta_{\min} = \frac{1}{1-p_{\min}}$ , and  $o_{\max} = \frac{p_{\max}}{1-p_{\max}}$ .

*Proof.*  $\frac{1}{1-x}$  and  $\frac{x}{1-x}$  are increasing functions for  $x \in (0, 1)$ . Therefore,

$$g_i(n) = \sum_{x \in \mathcal{X}} p_x^i (1-p_x)^{n-i} = \sum_{x \in \mathcal{X}} \left(\frac{1}{1-p_x}\right)^{-n} \left(\frac{p_x}{1-p_x}\right)^i \leq |\mathcal{X}| \cdot \beta_{\min}^{-n} o_{\max}^i.$$

□

**Theorem B.4.** The variance of the estimator  $\hat{M}_k^B$  is bounded as follows:

$$\text{Var}(\hat{M}_k^B) \leq c_1 \cdot n^{2k+1} \cdot c_2^{-n},$$

where  $c_1 = S \cdot \left(\frac{\epsilon}{k}\right)^{2k}$ ,  $c_2 = \min\left(\frac{1}{1-p_{\min}}, \frac{1-p_{\max}}{p_{\max}(1-p_{\min})}\right)$ . In other words,  $\text{Var}(\hat{M}_k^B) = \mathcal{O}(n^{2k+1} \cdot \beta_{\min}^{-n} \cdot \max(1, o_{\max}^n))$ .

*Proof.* From Lemma B.2 in the supplementary, we have  $\text{Var}(\Phi_i) \leq S f_i(n) - f_i(n)^2$ . Thus,

$$\begin{aligned}
 \text{Var}\left(\frac{\Phi_{k+i}}{\binom{n}{k+i}}\right) &\leq \frac{S f_{k+i}(n) - f_{k+i}(n)^2}{\binom{n}{k+i}^2} \leq S \cdot \frac{g_{k+i}(n)}{\binom{n}{k+i}} \leq \frac{S^2 \cdot \beta_{\min}^{-n} o_{\max}^{k+i}}{\binom{n}{k+i}} \quad (\text{by Lemma B.3}) \\
 \binom{n}{k}^2 \text{Var}\left(\frac{\Phi_{k+i}}{\binom{n}{k+i}}\right) &\leq S \beta_{\min}^{-n} \frac{\binom{n}{k}^2}{\binom{n}{k+i}} o_{\max}^{k+i} \leq S \beta_{\min}^{-n} \left(\frac{e^2 n^2}{k^2}\right)^k o_{\max}^{k+i} \leq S \beta_{\min}^{-n} \left(\frac{e^2 n(k+i)}{k^2}\right)^k o_{\max}^{k+i} \\
 &\leq S \beta_{\min}^{-n} \left(\frac{e^2 n^2}{k^2}\right)^k O_M = S \beta_{\min}^{-n} \left(\frac{en}{k}\right)^{2k} O_M, \quad \text{where } O_M = \max(o_{\max}^k, o_{\max}^n), \\
 \text{Var}(\hat{M}_k^B) &= \binom{n}{k}^2 \text{Var}\left(\sum_{i=1}^{n-k} (-1)^{i-1} \frac{\Phi_{k+1}}{\binom{n}{k+1}}\right) \\
 &\leq (n-k) \binom{n}{k}^2 \text{Var}\left(\frac{\Phi_{k+1}}{\binom{n}{k+1}}\right) \\
 &\leq S(n-k) \left(\frac{en}{k}\right)^{2k} \cdot \beta_{\min}^{-n} O_M = \mathcal{O}(n^{2k+1}) \cdot \beta_{\min}^{-n} O_M,
 \end{aligned} \tag{35}$$

where (35) follows from Cauchy-Schwarz inequality ( $\text{Var}(\sum_{j=1}^M X_j) \leq M \cdot \sum_{j=1}^M \text{Var}(X_j)$ ). The proof follows from dividing the variance of the estimator into two cases:  $o_{\max} < 1$  and  $o_{\max} > 1$ : If  $o_{\max} < 1$ ,  $O_M = o_{\max}^k$ , and  $\text{Var}(\hat{M}_k^B) = \mathcal{O}(n^{2k+1} \beta_{\min}^{-n})$ . If  $o_{\max} > 1$ ,  $O_M = o_{\max}^n$ , and,  $\text{Var}(\hat{M}_k^B) = \mathcal{O}(n^{2k+1} \beta_{\min}^{-n} o_{\max}^n)$ .  $\square$

Therefore, the variance exponentially decreases with  $n$  if  $p_{\max} < 0.5$  or  $\frac{1-p_{\max}}{p_{\max}(1-p_{\min})} < 1$ .

**Corollary B.5.** *There exists a constant  $c > 1$  such that*

$$\text{MSE}(\hat{M}_k^B) \leq \mathcal{O}(n^{2k+1} c^{-n}).$$

*Proof.* From Equ. (7) in the manuscript, the bias  $|\mathbb{E}(\hat{M}_k^B) - M_k| \leq S \cdot p_{\max} \cdot n^k \cdot p_{\max}^n$ . The proof follows from the fact that  $\text{MSE} = \text{Var} + \text{Bias}^2$  and the bound of the variance and the bias.  $\square$

### C. Computing the Variance and the MSE of the Evolved Estimators

Same as  $\hat{M}_k^B$ , the evolved estimators from the genetic algorithm are also linear combinations of  $\Phi_k(n)$ s (while varying both  $k$  and  $n$  unlike  $\hat{M}_k^B$ ). Given the evolved estimator  $\hat{M}_k^{\text{Evo}} = \sum_i c_i \Phi_{k_i}(n_i)$ , the expected value of  $\hat{M}_k^{\text{Evo}}$  is given by substituting  $\Phi_k(n)$  with  $f_k(n)$ :

$$\mathbb{E}(\hat{M}_k^{\text{Evo}}) = \sum_i c_i f_{k_i}(n_i). \tag{36}$$

Given the multinomial distribution  $p$ , the covariance between  $\Phi_k(n)$  and  $\Phi_{k'}(n')$ , which is needed to compute the variance of  $\hat{M}_k^{\text{Evo}}$  as Equ. (23), can be computed as follows:

**Theorem C.1.** *Given the multinomial distribution  $p = (p_1, \dots, p_S)$  with support size  $S$ , let  $X^{n_{\text{total}}}$  be the set of  $n_{\text{total}}$  samples from  $p$ . Let  $X^n$  and  $X^{n'}$  be the first  $n$  and  $n'$  samples from  $X^{n_{\text{total}}}$ , respectively; WLOG, we assume  $1 \leq n' \leq n \leq n_{\text{total}}$ . Then, the covariance of  $\Phi_k(n) = \Phi_k(X^n)$  and  $\Phi_{k'}(n') = \Phi_{k'}(X^{n'})$  ( $1 \leq k \leq n$ ,  $1 \leq k' \leq n'$ ) is given by following:*

$$\text{Cov}(\Phi_k(n), \Phi_{k'}(n')) = \mathbb{E}[\Phi_k(n) \cdot \Phi_{k'}(n')] - f_k(n) \cdot f_{k'}(n') \tag{37}$$

$$= \mathbb{E}\left[\left(\sum_x \mathbb{1}(N_x = k)\right) \cdot \left(\sum_{x'} \mathbb{1}(N'_{x'} = k')\right)\right] - f_k(n) \cdot f_{k'}(n') \tag{38}$$

$$= \sum_x \sum_{x'} \mathbb{E}[\mathbb{1}(N_x = k \wedge N'_{x'} = k')] - f_k(n) \cdot f_{k'}(n'), \tag{39}$$

where  $N'_{x'}$  is the number of occurrences of  $x'$  in  $X^{n'}$ . Depending on the values of  $n, n', k, k', x$ , and  $x'$ , the  $\mathbb{E}[\mathbb{1}(N_x = k \wedge N'_{x'} = k')]$  can be computed as follows:

$\forall n, n' \text{ s.t.}$	$\forall x, x' \text{ s.t.}$	$\forall k, k' \text{ s.t.}$	$\mathbb{E}[\mathbb{1}(N_x = k \wedge N'_{x'} = k')]$
$n = n'$	$x = x'$	$k = k'$	$\binom{n}{k} p_x^k (1 - p_x)^{n-k}$
		$k \neq k'$	0 (infeasible)
	$x \neq x'$	$k + k' \leq n$	$\frac{n!}{k!k'!(n-k-k')!} p_x^k p_{x'}^{k'} (1 - p_x - p_{x'})^{n-k-k'}$
		$k + k' > n$	0 (infeasible)
$n \neq n'$	$x = x'$	$k' \leq k$	$\binom{n'}{k'} p_x^{k'} (1 - p_x)^{n'-k'} \cdot \binom{n-n'}{k-k'} p_x^{k-k'} (1 - p_x)^{(n-n')-(k-k')}$
		$k' > k$	0 (infeasible)
	$x \neq x'$	$k + k' \leq n$	$\sum_{i=\max(0, k-(n-n'))}^{\min(k, n-k')} \frac{n!}{k'!i!(n'-k'-i)!} \frac{(n-n')!}{(k-i)!((n-n')-(k-i))!} p_x^{k'} p_{x'}^{k'} (1 - p_x - p_{x'})^{n'-k'-i} p_x^k (1 - p_x)^{(n-n')-(k-i)}$
		$k + k' > n$	0 (infeasible)

*Proof.* The proof is straightforward from the definition of  $N_x$  and  $N'_{x'}$ . □

Given the expected value and the variance of  $\hat{M}_k^{\text{Evo}}$ , the MSE of  $\hat{M}_k^{\text{Evo}}$  naturally follows.

## D. Details of the Hyperparameters of the Evolutionary Algorithm

For evaluating Algorithm 1, we use the following hyperparameters:

- Same as the Orlitsky's study (Orlitsky & Suresh, 2015), which assess the performance of the Good-Turing estimator, we use the hybrid estimator  $\hat{p}$  of the empirical estimate and the Good-Turing estimate to approximate the underlying distribution  $\{p_x\}_{x \in \mathcal{X}}$  for estimating the MSE of the evolved estimator. The hybrid estimator  $\hat{p}$  is defined as follows: If  $N_x = k$ ,

$$\hat{p}_x = \begin{cases} c \cdot \frac{k}{N} & \text{if } k < \Phi_{k+1}, \\ c \cdot \frac{\hat{M}_k^G}{\Phi_k} & \text{otherwise,} \end{cases}$$

where  $c$  is a normalization constant such that  $\sum_{x \in \mathcal{X}} \hat{p}_x = 1$ .

- The number of generations  $G = 100$ . To avoid the algorithm from converging to a local minimum, we limit the maximum number of generations to be 2000.
- The mutant size  $m = 40$ .
- When selecting the individuals for the mutation, we use *tournament* selection with tournament size  $t = 3$ , i.e., we randomly choose three individuals with replacement and select the best one, and repeat this process  $m$  times.
- When choosing the top three individuals when constructing the next generation, we use *elitist* selection, i.e., choosing the top three individuals with the smallest fitness values.
- To avoid the estimator from being too complex, we limit the maximum number of terms in the estimator to be 20.

The actual script implementing Algorithm 1 can be found at the publically available repository

<https://anonymous.4open.science/r/Better-Turing-157F>.

**E. Additional Experimental Results**

Table 2. The MSE of the Good-Turing estimator  $\hat{M}_0^G$ , minimal bias estimator  $\hat{M}_0^B$ , and the best evolved estimator  $\hat{M}_0^{\text{Evo}}$  and for the missing mass  $M_0$ , the success rate  $\hat{A}_{12}$  of the evolved estimator ( $X_2$ ) against the Good-Turing estimator ( $X_1$ ), and the ratio (Ratio,  $MSE(\hat{M}_0^{\text{Evo}})/MSE(\hat{M}_0^G)$ ) for two support sizes  $S = 100$  and  $200$ , three sample sizes  $n$  and six distributions.

$S$	$n/S$	Distribution	$MSE(\hat{M}_0^B)$	$MSE(\hat{M}_0^G)$	$MSE(\hat{M}_0^{\text{Evo}})$	$\hat{A}_{12}$	Ratio
100	0.5	uniform	6.834e-03	6.681e-03	6.267e-03	62%	93%
		half&half	6.821e-03	6.694e-03	4.489e-03	85%	67%
		zipf-0.5	6.676e-03	6.565e-03	3.311e-03	94%	50%
		zipf-1	4.995e-03	4.943e-03	3.065e-03	98%	62%
		diri-1	6.166e-03	6.086e-03	3.202e-03	96%	52%
		diri-0.5	5.223e-03	5.167e-03	2.708e-03	100%	52%
	1.0	uniform	2.365e-03	2.351e-03	1.905e-03	88%	81%
		half&half	2.200e-03	2.190e-03	1.439e-03	97%	65%
		zipf-0.5	2.207e-03	2.194e-03	1.982e-03	75%	90%
		zipf-1	1.713e-03	1.704e-03	1.705e-03	75%	100%
		diri-1	1.787e-03	1.778e-03	1.066e-03	100%	60%
		diri-0.5	1.388e-03	1.381e-03	8.747e-04	97%	63%
	2.0	uniform	4.047e-04	4.028e-04	3.428e-04	89%	85%
		half&half	4.237e-04	4.221e-04	3.035e-04	97%	71%
		zipf-0.5	4.580e-04	4.561e-04	3.321e-04	95%	72%
		zipf-1	4.826e-04	4.810e-04	3.633e-04	98%	75%
		diri-1	3.946e-04	3.932e-04	2.473e-04	100%	62%
		diri-0.5	3.276e-04	3.264e-04	2.587e-04	87%	79%
200	0.5	uniform	3.361e-03	3.323e-03	2.044e-03	95%	61%
		half&half	3.357e-03	3.326e-03	1.968e-03	96%	59%
		zipf-0.5	3.254e-03	3.227e-03	2.293e-03	89%	71%
		zipf-1	2.335e-03	2.324e-03	2.410e-03	74%	103%
		diri-1	3.011e-03	2.992e-03	2.359e-03	85%	78%
		diri-0.5	2.563e-03	2.550e-03	1.813e-03	94%	71%
	1.0	uniform	1.172e-03	1.169e-03	8.900e-04	99%	76%
		half&half	1.092e-03	1.090e-03	8.584e-04	99%	78%
		zipf-0.5	1.091e-03	1.088e-03	8.525e-04	97%	78%
		zipf-1	8.185e-04	8.165e-04	7.244e-04	88%	88%
		diri-1	8.898e-04	8.876e-04	6.652e-04	100%	74%
		diri-0.5	6.900e-04	6.882e-04	4.861e-04	98%	70%
	2.0	uniform	2.017e-04	2.012e-04	1.702e-04	93%	84%
		half&half	2.113e-04	2.109e-04	1.716e-04	100%	81%
		zipf-0.5	2.307e-04	2.302e-04	1.929e-04	100%	83%
		zipf-1	2.390e-04	2.387e-04	2.109e-04	96%	88%
		diri-1	1.961e-04	1.958e-04	1.648e-04	96%	84%
		diri-0.5	1.609e-04	1.607e-04	1.315e-04	93%	81%