Sequential Good-Turing and the Missing Species Problem

Oskar Andersson

Master's thesis
2022:E54

Lund University

Faculty of Science
Centre for Mathematical Sciences
Mathematical Statistics

Centrum Scientiarum Mathematicarum

Sequential Good-Turing and the Missing Species Problem

Oskar Andersson

June 27, 2022

Contents

1 Abstract

This essay introduces the sequential Good-Turing estimator and reviews the Good-Turing, Good-Toulmin and smoothed Good-Toulmin estimators. Some theoretical properties and drawbacks of the estimators are described. Monte-Carlo simulation is then used to compare the performance of the sequential Good-Turing estimator to the performance of the Good-Toulmin estimator along with the smoothed Good-Toulmin estimator, on both real and simulated data.

In certain scenarios the Monte-Carlo method outperforms the smoothed Good-Toulmin estimator.

2 Acknowledgements

Major thanks to Dragi Anevski whose help made this essay possible. A big thanks to students and staff at the department of mathematical statistics who helped me and supported me during my time in Lund.

3 Introduction

3.1 The basic problem

This essays concerns itself with two related estimation problems.

The first problem is this: You have been in a forest for a while and you have taken notes of how many animals you have seen. You have seen a lot of animals of some species and for some species you have only seen one or two specimen. You might then sit down on a stump and ponder, what's the probability of finding a new species if I observe one more animal?

According to the empirical frequency estimate, $\hat{p} = \frac{\text{number of previous observations}}{n}$ (where n is the sample size), the probability would be zero.

It seems reasonable that the estimate should be larger than 0, so how do we assign a positive probability to seeing something we have not observed before? And how do we then shift the remaining probability mass for the species that we have observed so that the normalization condition:

$$\sum_i \hat{p}_i \quad = \quad 1$$

is satisfied?

Although we borrow terminology from biology, similar problems exist in other fields where species are replaced by DNA-markers or words, for forensic science and linguistics, respectively. The problem also has applications in cryptography and one of the most common estimation methods was invented to aid in solving the Enigma during WWII.

During the work on the Enigma Alan Turing and John Good needed a way

to estimate the probability of encountering an unseen encryption key used by the Axis powers. The estimator they came up with is known as the Good-Turing estimator, cf. [5]. We will discuss this estimator more in depth in Section 4.2

There are quantities related to the discovery of new species other than probabilities that you might want to estimate, such as the *number* of new species discovered during a second sampling round. **This is known as the unseen species problem and is the second problem this essay is concerned with**.

More rigorously we have one data set of n animals and would like to say something about what number, U, of previously unseen species we would see if m additional animals have been appended to the original sample. Formally, we define

$$U \;\coloneqq\; \left|\left\{X_{n+1}^{n+m}\right\} \setminus \left\{X^n\right\}\right| \;,$$

where

$$\begin{aligned} \mathrm{X}_{n+1}^{m+n} &= [X_{n+1}, \ldots, X_{n+m}] \\ X^n &= [X_1, \ldots, X_n] \end{aligned}$$

In other words, the multiset X^n is a data set of size n and the multiset X_{n+1}^{n+m} is the data set when observing an additional m samples. Note that $|\,|$ denotes cardinality and $\{\ \}$ denotes a set. Note that X_{n+1}^{n+m} and X^n are a multisets while $\{X_{n+1}^{n+m}\}$ and $\{X^n\}$ are regular sets.

Usually, to estimate U an estimator called the Good-Toulmin estimator is used, this estimator will be discussed in section 2.3.

3.2 Previous research

Good-Turing estimation as well as other estimators such as the Pattern Maximum Likelihood-estimator, cf. [1], have been used to estimate the probability of drawing a previously unseen species.

Good-Toulmin estimation is used to estimate the number of new species seen during a new sampling round. Depending on the scenario, sometimes so called smoothing has to be used. There is little research that gives insight into which smoothing method is optimal. However, some asymptotic results exist, cf. [6].

3.3 Structure of essay

Section 4.1 describes the Good-Turing estimator, derives some theoretical properties and describes some drawbacks of the estimator together with potential fixes.

Section 4.2 describes the Good-Toulmin estimator and describes a modification (smoothing) of it that has to be used in certain scenarios.

Section 4.3 introduces an apparently novel sequential Good-Turing estimator which is a Monte-Carlo method that estimates the number of species discovered during a sampling round. This is the same entity that the Good-Toulmin estimator estimates.

A theoretical comparison between the true probabilites and those of the sequential Good-Turing estimator is then made.

Section 5 describes a simulation study which compares the performance of

sequential Good-Turing and the Good-Toulmin estimator.

Section 6 describes the results of the simulation study.

Section 7 discusses the results of the study and some of the advantages and disadvantages of sequential Good-Turing.

4 Theory

4.1 Probability of drawing an unseen species

We want to estimate C_r, the probability of drawing an animal that has been seen r times before. This is a sum of probabilities for all the species that have previously been seen r times.

$$C_r = \sum_{x \in S_r} \theta_x \tag{1}$$

where θ_x is the probability of drawing species x, and the index set S_r is the set of species that have been seen exactly r times before.

4.2 Good-Turing

One approach for estimating the probability of observing an unseen species is $\frac{\text{number of singletons}}{n}$, this is the estimate known as the Good-Turing estimate. A singleton is a species that has been observed only once and n denotes the size of the sample. A natural question that arises when using this estimate is how the probability mass for the seen species should be readjusted so that the probabilities sum to one. This is done by a generalization of the Good-Turing estimator presented above. To generalize the Good-Turing estimator

we must first introduce some notation. Let

$$Z = (z_1, z_2, ..., z_n)$$

be the data we have collected, consisting of symbols (animals) that can be classified into different species. In this data set there are likely symbols that occur more than once.

Let S_r be the set of species that occur r times, and define

$$\tilde{N}_r \quad = \quad |S_r|$$

the number of species with r animal observations, for $r = 1, 2, ...$ Thus $\tilde{N}_0$ is the number of unseen species, $\tilde{N}_1$ is the number of singletons, $\tilde{N}_2$ is the number of species with two observed animals, etc.

The Good-Turing estimator of observing a species that has previously been observed r times is defined as

$$\hat{p_r} \quad = \quad \frac{(r+1)\tilde{N}_{r+1}}{n} \tag{2}$$

Let x denote a species label and let $\theta = \theta_1, \theta_2, ...$ be the unknown population frequencies in nature, i.e. θ_i is the probability of seeing an animal of species i when sampling one animal from nature, for $i = 1, 2, ...$

As hinted at in *Section 2.1* we would like to adjust the relative frequency we saw in our data so that we shave off a little bit of probability from the observed events and give it to the unobserved events. The Good-Turing estimator does exactly this.

Now that we have introduced the Good-Turing estimator we will revisit the

quantities we want to estimate and try to formulate the problem into a more treatable form.

Recall that S_r is the set of species seen r times and define the total probability of seeing those species seen r times as

$$\begin{aligned} C_r &= \sum_{x \in S_r} \theta_x \\ &= \sum_{x} \theta_x 1_r(x) \ , \end{aligned} \tag{3}$$

and with $1_r(x)$ being the indicator for observing species x an r amount of times.

What is particular about the types of problems dealt with in this thesis is that we are interested in drawing inference about things related to S_0, a set that we have not observed.

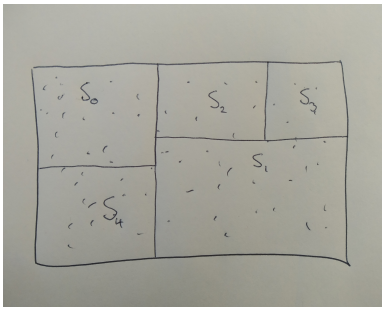

Figure 1: All animals divided up into sets depending on how often they have been observed.

We will use the sets S_1 to draw inference about S_0. This is the way we have to go about things since by definition we cannot observe S_0. In the same fashion we will use S_2 to draw inference about S_1, S_3 to draw inference about S_2 etc.

This estimation problem is different from the typical case in that the quantity we are trying to estimate is actually dependent on the sample, namely the probability of seeing a species not previously observed.

We may compare the probability of seeing an unseen species with ordinary parameter estimation where we only gain information about the parameter when we get additional data but the parameter stays fixed. When estimating the probability of seeing an unseen species the quantity we are trying to estimate actually changes when we get new data.

4.2.1 Good-Turing Example

When sampling words from *The Great Gatsby* you might end up with data that looks like Z, below.

$$Z \;=\; \{\text{the, green, light, Daisy, egg, west, the, and, jazz, the, and, a, a }\},$$

which would give the observed word labels

$$X \;=\; \{\text{the,green, light, Daisy, egg, west, and, jazz,a}\}.$$

The sample would give

$\mathrm{S}_1 = \{green, light, Daisy, egg, west, jazz\}$

$S_2 = \{and, a\}$

$S_3 = \{the\}$

$\tilde{N}_1 = 6$

$\tilde{N}_2 = 2$

$\tilde{N}_3 = 1$

We want to estimate the probability of seeing an unseen word if we look up a word in the book at random. The Good-Turing estimate of observing a previously unseen word when drawing one more would be:

$$\hat{p}_0 = \frac{6}{13}$$

Similarly, the Good-Turing estimators of p_1 and p_2 are

$$\hat{p}_1 = \frac{(1+1)2}{13} = \frac{4}{13}$$
$$\hat{p}_2 = \frac{(1+2)1}{13} = \frac{3}{13}$$

Note in particular that $\hat{p}_3 = 0$. This is a consequence of the sequential way we estimate the probabilities. In some sense this is an undesirable outcome since it seems sensible that the event that a word occurs three times should have a non-zero probability, and especially since we have actually observed the event before.

Peculiarities like this, as well as other problems are usually solved by smoothing the data, which will be discussed in Section 2.2.3.

4.2.2 Good-Turing intuition

In order to illustrate the intuition behind the Good-Turing estimator, we re-phrase the estimation problem slightly.

Given that you have collected data, what is the probability of seeing an animal for the $(r+1)'th$ time?

One approach is to just say that it is the same as the relative frequency of animals seen $r+1$ times. Namely,

$$\hat{p}_r = \frac{(r+1)\tilde{N}_{r+1}}{n}$$

Note that the estimate of seeing something unseen for the first time would be given by the above formula with $r = 0$.

Judging from the formula for the Good-Turing estimator you can see that when estimating p_r we "leak" some probability from those species that have been observed one more time than those in S_r in a sequential fashion.

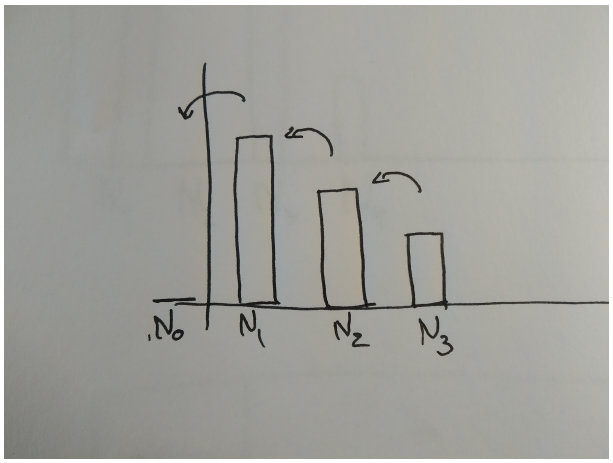

Figure 2: The probability of seeing something from S_r is estimated with the help of $\tilde{N}_{r+1}$.

4.2.3 Smoothing

There exists a potentially fatal flaw with the version of the Good-Turing estimator presented above, if

$$\tilde{N}_{r+1} = 0 \ .$$

The estimator would then assign zero probability to the event of observing a species (or word) r times, which may be undesirable since, we could very

well have data such that $\tilde{N}_r > 0$. Of course $\tilde{N}_r > 0$ means that the event of observing a species r times has happened before.

In real data these gaps where some $\tilde{N}_r = 0$ occur more often for large r but it could also be that you have no observations of singletons, meaning that the Good-Turing estimator fails in its original mission of estimating the probability of observing an unseen species.

Another problem that is perhaps more relevant to this essay is that you can end up in the situation that $\hat{p}_r > 0$ and $\tilde{N}_{r-1} = 0$ This means that if we go out and collect one more data point, the estimator suggests that there is a possibility that the new data point is an animal seen for the $r'th$ time. Clearly, this is impossible because we would have to have seen it $r - 1$ times before collecting the additional data point, and obviously we have not since $\tilde{N}_{r-1} = 0$. It seems like an undesirable property that the estimator assigns positive probability to events that cannot happen. Note that this property is not only undesirable from a logical point of view but also presents real issues when doing e.g. Monte Carlo simulations.

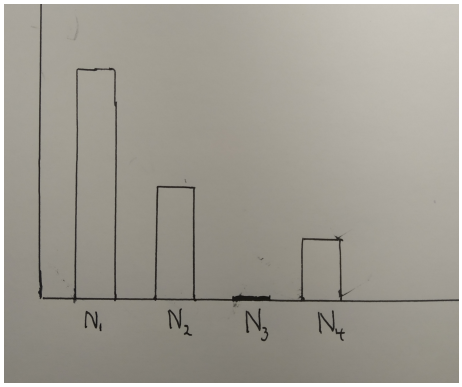

Figure 3: Gaps in the $\tilde{N}_r$'s lead to issues for the Good-Turing estimator.

There seems to be two main solutions to the problems above: Smooth the data or use different estimators on different parts of the data. Usually the gaps in the $\tilde{N}_r$'s occur for higher values of r. This implies that you can often use the Good-Turing estimate when estimating θ_r for low r and e.g. a power law for higher r, cf. [3]. To introduce smoothing, define

$$Z_r \;=\; \frac{\tilde{N}_r}{0.5(t-q)}\ ,$$

where t, q and r are consecutive subscripts where $\tilde{N}_t$,$\tilde{N}_q$, $\tilde{N}_r$ are all non zero.

One suggested smoothing procedure is to do linear regression between some quantities related to the $\tilde{N}_r$'s and then use the regression line instead of the observed values. The suggested quantities are Z_r and r. In the linear regression analysis $log(Z_r)$ is treated as the dependent variable and $log(r)$ is the explanatory variable.

For the purposes of this essay a different smoothing procedure will be used. The reason for this is our use of a Monte-Carlo scheme that requires us to sequentially change the $\tilde{N}_r$'s and we want them to remain integers which the above scheme does not guarantee.

4.2.4 Poisson sampling

Suppose x_1^n is the sample drawn from $(\theta_1, \theta_2, ...)$, and suppose N_x is the number of times that species x appears in x_1^n.

Poisson sampling means that we

1. Generate a random number $N \in Po(n)$

2. Generate $x_1, ..., x_n$ from the distribution $(\theta_0, \theta_1, ..., \theta_n)$

Then a standard result is ,cf. [2], that if N_x is the number of times x appears in the above sampling then

a) $N_x \in Po(n\theta_x)$

b) The N_x are independent.

4.2.5 Unbiasedness of the Good-Turing estimator

In a later part of the essay we will use the unbiasedness of the Good-Turing estimate to show some theoretical properties of our suggested scheme. The proof below also gives some insight into why Good-Turing works.

In the following proof we will show unbiasedness for Poisson sampling. If we make looser assumptions than Poisson sampling we can instead show that the bias is small, cf. [4].

Consider the sum of probabilities of all species that appear r times:

$$C_r = \sum_x \theta_x 1_r(x) \ .$$

The Good-Turing estimate of seeing an animal of a species that has been seen r times before is

$$\begin{aligned} \hat{p}_r &= \frac{r+1}{n} \cdot \tilde{N}_{r+1} \\ &= \frac{r+1}{n} \cdot \sum_x 1\{x \in S_{r+1}\} \ . \end{aligned}$$

Taking expectation one gets

$$\begin{aligned} E[\hat{p}_r] &= \sum_x \frac{r+1}{n} \cdot E[1\{x \in S_{r+1}\}] \\ &= \sum_x \frac{r+1}{n} \cdot P\{\text{species } x \text{ appears r+1 times}\} \\ &= \sum_x \frac{r+1}{n} e^{-\lambda_x} \cdot \frac{(\lambda_x)^{r+1}}{(r+1)!} \quad (\lambda_x = n\theta_x) \\ &= \sum_x \frac{\lambda_x}{n} \cdot e^{-\lambda_x} \cdot \frac{(\lambda_x)^r}{r!} \\ &= \sum_x \theta_x \cdot E[1\{x \in S_r\}] \\ &= E[C_r] \end{aligned}$$

where the third equality follows by property a) in Section 4.2.4, and the last equality follows by (3).

4.3 Good-Toulmin

As mentioned earlier one might be interested in knowing the number of new species one would discover when going out into nature and sampling animals. Clearly this is not only dependent on the pool of animals you are sampling from but also how many samples you take.

The *number* of unseen species can be estimated by the Good-Toulmin estimator, defined as

$$U^{GT} = \sum_{i=1}^{\infty} (-t)^i \tilde{N}_i \quad , \tag{4}$$

where $t = \frac{m}{n}$ is the ratio between the sample size m of the new sample and the sample size n of the old sample.

Thus we have a data set of size n and we would like to make inference about how many new species we would see if we went out into nature and took a sample of size m. Note that we do not actually go out and collect this second data set. We make inference about what the outcome would be if we would collect the second data set, and the inference is based on the first data set and the fact that we know that the second data set *would* be of size m.

The Good-Toulmin estimator is not a good estimator in all situations. If $t > 1$ then t^i grows exponentially, while the actual U grows at most linearly in t. The terms in the end of the sum in (3) grow very fast because of the exponential growth of the factor t and this will increase the variance of the estimator, cf. [6]. This effect is quite strong, meaning that most estimates with the Good-Toulmin estimator where $t > 1$ are very bad.

One approach to try to correct this is to truncate the sum at some location. The problem is that this gives a lot of weight to the last term in the sum and the estimates will still be quite bad. A clever solution to this problem is to use the smoothed Good Toulmin estimator, defined as

$$U^{\mathrm{L}} \;=\; E_L\left[-\sum_{i=1}^{L}(-t)^i \tilde{N}_i\right] \;.$$

Here L is some random truncation. One can think of this as generating a bunch of stopping times, L, from some distribution and taking the theoretical average of the truncated Good Toulmin estimator for these stopping times. The alternating sign property of the sum ensures that the problem with the high weight on the last term is mitigated.

Simplifying the expression for the Good-Toulmin estimator

$$\begin{aligned} U^L &= \mathrm{E}_L\left[-\sum_{i\geq 1}(-t)^i \tilde{N}_i \mathbf{1}_{i\leq L}\right] \\ &= -\sum_{i\geq 1}(-t)^i \,\mathrm{P}(L\geq i)\tilde{N}_i \ . \end{aligned}$$

Thus we end up with a weighted version of the Good-Toulmin estimator. What distribution and parameters L should have is unclear. Previous results state asymptotic error bounds for binomial and Poisson smoothing when certain parameters are picked, cf. [6].

The version of the smoothed Good-Toulmin estimator that will be used in this essay is the $Binomial(k,q)$ distribution for L with parameters;

$$k = \left\lfloor \frac{1}{2}\log_3 \frac{nt^2}{t-1} \right\rfloor, q = \frac{2}{t+2} \ ,$$

where $\lfloor\ \rfloor$ denotes the floor function.

With this choice one obtains the following asymptotic bound of the error:

$$\begin{aligned} \mathcal{E}_{n,t}\left(U^{\mathrm{L}}\right) &< n^{-\log_3(1+2/t)} \ , \\ \mathcal{E}_{n,t}\left(U^{\mathrm{L}}\right) &= U - U^L \ , \end{aligned}$$

4.3.1 Derivation of Good-Toulmin

Let N_x denote the number of times that species x has been observed in the first size n sample and let N'_x denote the number of times species x has been observed in the second size m sample. Then

$$U = \sum_x 1\left\{N_x = 0\right\} \cdot 1\left\{N'_x > 0\right\} \ .$$

Then under Poisson sampling by property a) and since N_x and N'_x are independent (for the same species), we obtain

$$\begin{aligned}
E(U) &= \sum_x E\left(1\left\{N_x = 0\right\}\right) E\left(1\left\{N'_x > 0\right\}\right) \\
&= \sum e^{-\lambda_x}\left(1 - e^{-\lambda'_x}\right) \qquad (\lambda_x = n\theta_x, \lambda'_x = m\theta_x) \\
&= \sum_x e^{-\lambda_x}\left(1 - e^{-t\lambda_x}\right) \\
&= -\sum_x e^{-\lambda_x} \cdot \sum_{i=1}^{\infty} \frac{\left(-t\lambda_x\right)^i}{i!} = -\sum_{i=1}^{\infty}(-t)^i \cdot \sum_x e^{-\lambda_x}\frac{\lambda_x^i}{i!} \\
&= -\sum_{i=1}^{\infty}(-t)^i \cdot E\left[\tilde{N}_i\right] \\
&= E\left[U^{\mathrm{GT}}\right] \ .
\end{aligned}$$

By the above argument we see that for Poisson sampling the Good Toulmin estimator is an unbiased estimator of the expected number of new species you will discover when taking another sample.

4.3.2 Good Toulmin Example

Using the same data as in the Good-Turing example and wanting to estimate the number of new species when sampling four more data points: $t = 4/13$
$\tilde{N}_1 = 6$,
$\tilde{N}_2 = 2$,
$\tilde{N}_3 = 1$, the Good-Toulmin estimator gives the following estimate of the expected number of previously unseen species.

$$U^{GT} = (\frac{4}{13})6 - (\frac{4}{13})^2 2 + (\frac{4}{13})^3 \approx 1.7 \ .$$

Note that if we want to sample 55 new data points then the estimator predicts that we would find approximately 65 new species,

$$\frac{55}{13}6 - (\frac{55}{13})^2 2 + (\frac{55}{13})^3 \approx 65 ,$$

which of course is not possible, this result is a consequence of t being quite high. The effect is even more pronounced if you have more data since you likely have t^i terms with a large i.

4.4 Sequential Good-Turing

This essay has reviewed the Good-Toulmin estimator as a way to estimate the number of discovered species during a new sampling round. One alternative approach is to use Monte-Carlo methods in conjunction with the Good-Turing probability estimate.

We use $\mathcal{F}_n$ to denote the information available at time step n. We use superscripts to denote what different quantities are in time, e.g. $\tilde{N}_r^n$ and $\tilde{N}_r^{n+m}$. This notation is needed since we know what has happened during the first n samples, and we are interested in estimating what will happen during a second sample of size m.

The main idea of sequential Good-Turing is to simulate sampling one new data point by simulating a data point using the Good-Turing estimate. Then you can append this simulated data point to your real data and use all of this data in your new Good-Turing estimates. By doing this repeatedly you can estimate the number of new species you would discover during a second sampling round.

More concretely, the algorithm is as follows: Given as sample of size n

(i) Estimate the Good-Turing probabilities: $\hat{p}_0, \hat{p}_1, ..., \hat{p}_k$

(ii) Generate a new data point: $Y \sim multinomial(1, \hat{p}_0, \hat{p}_1, ..., \hat{p}_k)$

(iii) Update the $\tilde{N}_i$'s

(iv) Repeat.

If the species Y has been seen $r-1$ times before and is now seen for the r'th time $\tilde{N}_r^{n+1}$ is added to and one is added which is clear but at the same time we have to keep in mind that a species previously seen $r-1$ times is now seen r times and therefore we have to subtract one from $\tilde{N}_{r-1}^n$. This subtraction is however not needed when we see a species for the first time.

Thus, if Y has been seen $r-1$ times before and is now seen again, for the $r'th$ time, the counts are updated according to

$\tilde{N}_r^{n+1} = \tilde{N}_r^n + 1$

$\tilde{N}_{r-1}^{n+1} = \tilde{N}_{r-1}^n - 1$

unless r = 1, then the $\tilde{N}_i$'s are updated according to:

$\tilde{N}_1^{n+1} = \tilde{N}_1^n + 1$.

When doing this the $\tilde{N}_i$'s behave as if we are iteratively sampling from the true data set. Of course, we are only estimating the true probabilities so this algorithm only approximates what would happen if we went out and sampled from the real population.

We can go through the algorithm m times and count the number of times a

previously unseen species comes up. If we do this a large number of times and record the average outcome we have an alternative way to estimate the quantity U.

The reason for adding one to zero counts is because of what was put forth in the *smoothing* section. Traditionally other smoothing techniques are used for Good-Turing but because of the discrete nature of our algorithm we need smoothing that results in discrete data.

Note that in the sampling scheme we have counted new species in a sequential way where we check for new species each step n, $n+1$, $n+2$.. until m. Whereas in real life we count the number of new species after we have the second data set of size m. The two ways of counting previously unseen species are equivalent.

The smoothing problem of Good-Turing estimation usually only concerns itself with smoothing out the probabilities but here we change the $\tilde{N}_i$'s and need to make sure that the algorithm does not subtract one from an $\tilde{N}_i$ that is equal to zero.

This Monte Carlo scheme could potentially be expanded to one where you record the behaviour of each species. Such an expansion would require more computational power and the smoothing problem is likely to be even more worrisome for this setup. In this thesis, this more advanced sampling is not performed.

4.5 Drawing one more animal, expectation of sequential Good-Turing and the behaviour of the combined probabilities

This section clarifies how the probability of drawing an animal that has been seen r times behaves when sampling from the population. There is also a derivation of the expectation of the Good-Turing estimate when using a hybrid data set consisting of both real and simulated data. The main reason for this is to check that the estimated probabilities do not deviate too much from the true ones.

4.5.1 Behaviour of combined probability

I use the superscripts *n+1* and n to denote quantities before and after adding the data point X, where X is a species label for an animal drawn from the population.

The quantity we are interested in estimating is:

$$C_r^{n+1} \quad = \quad \sum_{x \in S_r^{n+1}} \theta_x \quad ,$$

where S_r^{n+1} might have one element more or less than S_r^n due to the addition of X to the data.

After drawing a new species label we do not necessarily need to know the name of the species to know how the sum of probabilities behave, we just need to know that for some r we have that $S_r^{n+1} \ni X$.

For this to happen we need to have that $X \in S_{r-1}^n$. If X is an arbitrary

element in S^n_{r-1} then the probability of drawing X is just the sum of probabilities of every element in S^n_{r-1}, i.e.

$$\begin{aligned} P(X \in S^{n+1}_r) &= \sum_{x \in S^n_{r-1}} \theta_x \\ &= C^n_{r-1} \ , \end{aligned}$$

$$\begin{aligned} P(X \in S^n_r) &= \sum_{x \in S^n_r} \theta_x \\ &= C^n_r \ . \end{aligned}$$

How would drawing one more animal affect C^n_r? Well, there are three scenarios but first let us go over some assumptions. The first assumption is that drawing a new animal does not affect the values of any of the population probabilities, θ_x. This is a reasonable assumption if the data set is sufficiently large or if we return the sampled animals to the population. Let us also assume that each individual animal of the same species has the same probability of being drawn.

Scenario 1: We see an animal for the $r'th$ time, i.e. we get one more observation of one of the species previously seen $r-1$ times. Then

$$X \in S^{n+1}_r \ ,$$

$$C^n_r = \sum_x \theta_x 1\{x \in S_r\}$$

gets an additional term and is updated to

$$C^{n+1}_r = C^n_r + \theta_x \ ,$$

where θ_x is the probability of drawing X.

Scenario 2: We see something for the $r+1$'th time and therefore one term is subtracted from C_r

$$X \in S_{r+1}^{n+1} \ .$$

Scenario 3: Nothing happens to C_r, when a new animal is drawn, i.e.

$$X \notin S_{r+1}^{n+1} \cup S_r^{n+1}$$

This does not affect C_r.

4.5.2 Expectation of sequential Good-Turing.

We want to investigate the theoretical properties of the scheme described in *Section 4.4*. We investigate the expectation of the sequential Good-Turing estimator during addition of one simulated data point, to ascertain wether it is a reasonable estimator.

The Good-Turing estimator is:

$$\hat{p}_r^n \ = \ \frac{(r+1)\tilde{N}_{r+1}^n}{n} \ . \tag{5}$$

After simulating one data point, Y, and appending it to our real data the estimator turns into:

$$\hat{p}_r^{n+1} \ = \ \frac{(r+1)\tilde{N}_{r+1}^{n+1}}{n+1} \ .$$

Note that:

$$\tilde{N}_r^{n+1} \ = \ \tilde{N}_r^n + 1\{Y \in S_{r-1}^n\} - 1\{Y \in S_r^n\} \ . \tag{6}$$

The indicator functions are due to the fact that elements can both enter and leave S_r depending on which index set, S_r or S_{r-1}, the simulated data point Y belongs to. If we want to be strict about things we have not actually seen the data points before since we are just simulating from a distribution that is approximately how likely we are to draw and Y from the different S_r's.

Note the difference in our Monte-Carlo scenario from how the C_r's behave when we draw another sample, from the real population. In our Monte-Carlo scenario Y is drawn using the estimated probabilities.

From (7) and using that Y is simulated from the estimated Good-Turing probabilities at time n we get

$$E\left[\tilde{N}_{r+1}^{n+1} \mid \mathcal{F}_n\right] = \tilde{N}_{r+1}^{n} + \hat{p}_{r-1}^{n} - \hat{p}_{r}^{n} \quad . \tag{7}$$

We will use this expression for the study of the sequential Good-Turing estimator. Taking the conditional expectation of (6), and with the use of (8), we obtain

$$\begin{aligned} E\left[\hat{p}_r^{n+1} \mid \mathcal{F}_n\right] &= \frac{(r+1)}{n+1}[\tilde{N}_{r+1}^{n+1} \mid \mathcal{F}_n] \\ &= \frac{(r+1)}{n+1}\left(\tilde{N}_{r+1}^{n} + \hat{p}_{r-1}^{n} - \hat{p}_{r}^{n}\right) \\ &= \frac{n}{n}\frac{(r+1)}{n+1}\left(\tilde{N}_{r+1}^{n} + \hat{p}_{r-1}^{n} - \hat{p}_{r}^{n}\right) \\ &= \frac{n}{n+1}\hat{p}_r + \frac{(r+1)}{n+1}\left(\hat{p}_{r-1}^{n} - \hat{p}_{r}^{n}\right)) \end{aligned}$$

We have previously shown that under Poisson sampling the Good-Turing estimate is unbiased.

We suggest the following conjecture

Conjecture 4.1 *Under Poisson sampling, sequential Good-Turing estimator satisfies.*

$$E\left[\hat{p}_r^{n+1} \mid \mathcal{F}_n\right] \;=\; E[\frac{n}{n+1}C_r^n + \frac{(r+1)}{n+1}\left(C_{r-1}^n - C_r^n\right)]$$

4.5.3 Expectation of C_r when adding data points

We now investigate the expectation how C_r behaves as we observe a new data point.

$$C_r^{n+1} \;=\; C_r^n + \sum_{g \in S_{r-1}^n} \theta_g 1[X=g] - \sum_{g \in S_r^n} \theta_g 1[X=g] \; .$$

Then taking the conditional expectation of the expression we obtain

$$\begin{aligned}
E[C_r^{n+1} \mid \mathcal{F}_n] &= C_r^n + \sum_{g \in S_{r-1}^n} \theta_g E[1[X=g]] - \sum_{g \in S_r^n} \theta_g E[1[X=g] \\
&= C_r^n + \sum_{g \in S_{r-1}^n} \theta_g P[X=g] - \sum_{g \in S_r^n} \theta_g P[X=g] \\
&= C_r^n + \sum_{g \in S_{r-1}^n} \theta_g^2 - \sum_{g \in S_r^n} \theta_g^2 \\
&= \sum_{g \in S_r^n} \theta_g + \sum_{g \in S_{r-1}^n} \theta_g^2 - \sum_{g \in S_r^n} \theta_g^2 \\
&= \sum_{g \in S_r^n} (1-\theta_g)\theta_g + \sum_{g \in S_{r-1}^n} \theta_g \cdot \theta_g
\end{aligned}$$

The attentive reader might note that this "proof of reasonableness" of the sequential Good-Turing estimator does not take into consideration the special case where $r = 0$.

The proof also only shows what happens when doing the first step in the Monte Carlo scheme. It is possible that the error accumulates when one takes several steps.

5 Monte-Carlo simulation and real data analysis

5.1 Comparison and simulation

Now that we have described the different estimators and looked at some theoretical properties we would like to see how the estimators perform on data. I use word occurences from two different books as data as well as simulated data from a Zipf distribution with a couple of different parameters.

The Zipf distribution has probability mass function:

$$f(x) = \frac{1}{x^\alpha \sum_{i=1}^{n} (1/i)^\alpha}, \quad x = 1, 2, \ldots, n$$

The reason for picking these data sets is that one relatively common use for Good-Toulmin is drawing inference about linguistic data. Word occurences in books and corpuses approximately follow a Zipf distribution, cf. [7]

This section explains where the data comes from and what sort of comparison is used to evaluate the performance of the estimators.

There are three estimators used in this section: Sequential Good-Turing, Good-Toulmin and Binomially Smoothed Good-Toulmin. The smoothing parameters used are those given in Section 4.3.

5.1.1 True number of discovered species

For a general data set we do not have theoretical results for how many new species we will see during the second sampling round. So how do we find a quantity to compare the estimators with?

Our approach is to use Monte-Carlo methods to estimate the quantity. The method mimics what happens during real sampling. First a data set is generated and a first subsample is randomly picked out. Then a second subsample is picked out from the data and the number of species is compared to that of the first subsample. Each individual animal is sampled with equal weight.

An alternative approach is to let m be the number of remaining data points in the original data set. In this case U can be computed without having to resort to Monte Carlo.

5.1.2 About the data

All inflections and conjugations of words are treated as different words. This means that two words only count as the same if they are lexicographically identical, save for capital letters.

The following punctuations marks have been removed from the data: !,?,',". .

5.1.3 Loss function and simulated data

The loss function used to compare the different estimators is:

$$L = \left(\frac{\hat{U} - U}{m} \right)^2 ,$$

where U is the true number of previously unseen (now discovered) species and $\hat{U}$ is the estimated number of previously unseen (now discovered) species.

Note that the error is "normalized" by the size m of the second sample. We use this normalization because the maximum number of discovered species

grows with the size of the second sample.

The reason for using the Zipf distribution is that this is a distribution often occuring for word occurences in quantitative linguistics. It is therefore natural to use next to to real linguistic data.

The real linguistic data used in this thesis comes from F. Scott Fitzgerald's 1925 book *The Great Gatsby* and Oscar Wilde's 1891 book *The Picture of Dorian Gray*. The Great Gatsby contains 46688 words and The Picture of Dorian Gray contains 74869 words. The simulated data sets all contain 1000 data points.

5.2 Comparison Scheme

The estimators are compared as follows:

1. Generate some data or use real data.
2. Pick out a subsample from the data of length q.
3. Do a Monte-Carlo simulation and record how many new species you will see if you sample m more data points, provided you have already seen the subsample in the previous step.
4. Record the sequential Good-Turing estimate and the Good-Toulmin estimate of how many unseen species you would see if you drew m more samples, using the subsample already picked out.
5. Compare the estimates to the "pure Monte-Carlo" result.
6. Repeat and record averages of the loss function.

For all of the results in this essay we use 1000 Monte-Carlo samples for sequential Good-Turing and 25 rounds for the Monte-Carlo Scheme which compares the estimators to the true number of discovered species.

5.2.1 Comparison without using Monte-Carlo

It is possible to come up with a number of species you will discover when sampling provided that you exhaust the data set. For computational reasons I use data of size 100. Granted this is a situation very different from what the estimators are used for I choose to include in my essay as an alternative way to evaluate the performance of the estimators.

6 Results

6.1 Comparison with ordinary Good-Toulmin

For virtually all situations where Good-Toulmin was applicable, meaning where $t \leq 1$, it outperformed sequential Good-Turing. Recall that $t = \frac{m}{n}$ where n is the size of first sample and m is the size of the second sample. Refer to the appendix for more details on the results.

6.2 Comparison with smoothed Good Toulmin

For situations where $t > 1$ and we therefore have to use a smoothed version of the Good-Toulmin estimator, sequential Good-Turing seems to outperform the Good-Toulmin estimator. We used binomial smoothing and the smoothing parameters suggested in Orlitsky et al. [6].

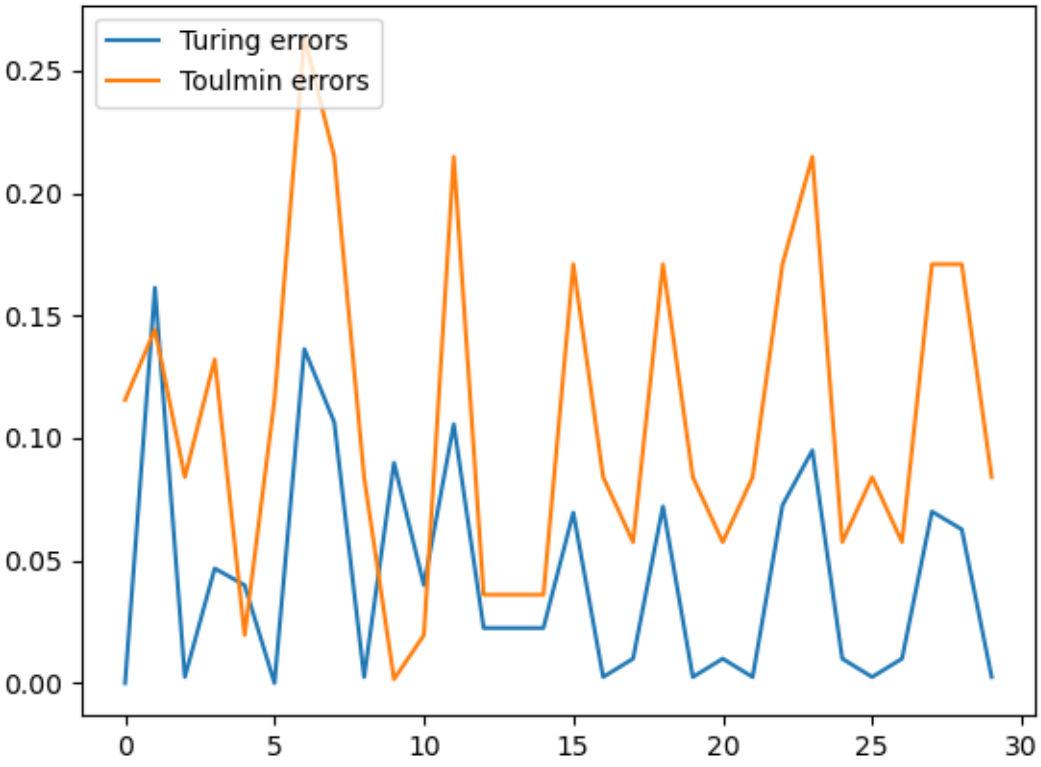

Figure 4: Graph of loss for the two estimators.

When using data simulated from a Zipf distribution the two estimators seem to be more or less equally good although sequential Good-Turing might be a little better.

7 Discussion

For the data used in this thesis sequential Good-Turing showed no increase in performance compared to ordinary Good-Toulmin. If anything, it was a little bit worse.

Where sequential Good-Turing shows promise is for real data when $t > 1$ and therefore smoothed Good-Toulmin has to be used. For some reason this is not the case for simulated data.

It could be that the parameters we used for the smoothed Good-Toulmin are suboptimal and therefore sequential Good-Turing outperforms. Possibly due to outliers in the real data.

This in and of itself shows that there is an advantage in sequential Good-Turing as there is no need to specify any parameters and the tools for deciding parameters in the smoothed Good-Toulmin estimator are limited.

The smoothing method used for the sequential Good-Turing method was based on practical considerations rather than theoretical justifications. Another smoothing method might very well improve the performance of the Good-Turing estimator.

This essay only very briefly investigates the theoretical properties of the sequential Good-Turing estimator. The expectation for the sequential Good-

Turing is compared with the true probabilities. It would be valuable if there was a theoretical comparison between sequential Good-Turing and the expected value of U.

Another point to consider is that the Good-Toulmin estimator requires virtually no computational power while the sequential Good-Turing estimator can be computationally costly if m is big.

Further research could involve Monte-Carlo methods with different estimators than the Good-Turing one, such as pattern maximum likelihood.

References

[1] J. Acharya, H. Das, A. Orlitsky, and A. T. Suresh. A unified maximum likelihood approach for optimal distribution property estimation. *CoRR*, abs/1611.02960, 2016.

[2] B. Efron and R. Thisted. Estimating the number of unsen species: How many words did shakespeare know? *Biometrika*, 63(3):435–447, 1976.

[3] W. A. Gale. Good-turing smoothing without tears. *Journal of Quantitative Linguistics*, 2, 1995.

[4] D. A. McAllester and R. E. Schapire. On the convergence rate of good-turing estimators. In *Proceedings of the Thirteenth Annual Conference on Computational Learning Theory*, COLT '00, page 1–6, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[5] A. Orlitsky, N. P. Santhanam, and J. Zhang. Always good turing: Asymptotically optimal probability estimation. *Science*, 302(5644):427–431, 2003.

[6] A. Orlitsky, A. T. Suresh, and Y. Wu. Optimal prediction of the number of unseen species. *Proceedings of the National Academy of Sciences*, 113(47):13283–13288, 2016.

[7] A. Ullah. page 138–140. CRC Press, 2011.

8 Appendix

The appendix contains tables for the results of the comparison schemes described in Section 5.2. Note that sequential Good-Turing is compared with both the ordinary and smoothed Good-Toulmin estimator. The smoothed Good-Toulmin estimator is used whenever the ratio $\frac{n}{m} > 1$. n denotes the size of the first data sample and m denotes the size of the second data sample.

n	m	Turing loss	Smoothed Toulmin loss
10	20	0.017 39	0.019 69
10	15	0.035 81	0.024 00
10	11	0.014 86	0.027 89
10	25	0.014 37	0.016 09
10	30	0.006 59	0.011 54
20	25	0.003 80	0.010 71
20	30	0.002 86	0.013 18
20	35	0.002 91	0.006 66
20	40	0.003 22	0.007 31

Table 1: Smoothed Good Toulmin and Sequential Good-Turing evaluated on simulated Zipf(2) distributed data

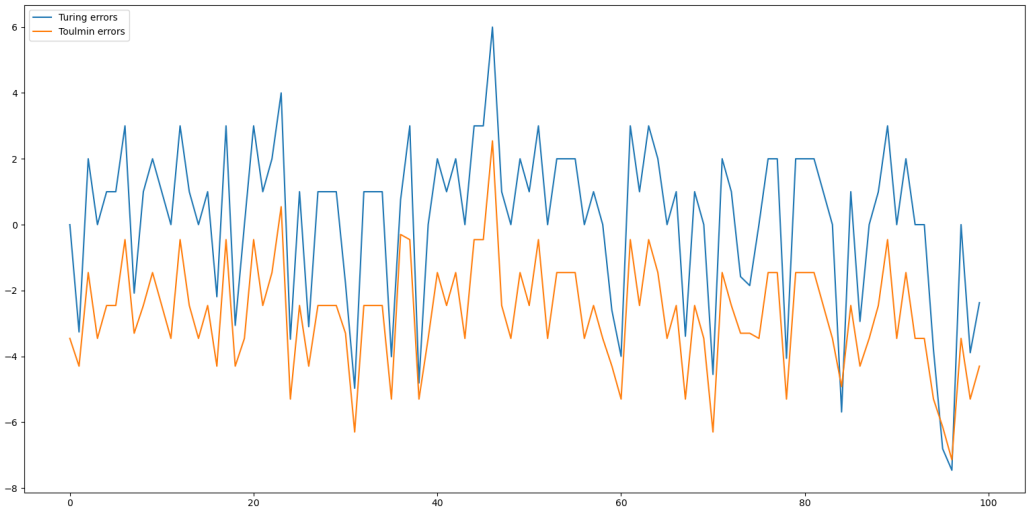

Figure 5: Graph of errors for the two estimators.

n	m	Turing loss	Toulmin loss
10	20	0.005 27	0.005 08
10	15	0.007 89	0.008 21
10	11	0.008 79	0.009 33
10	25	0.007 42	0.007 98
10	30	0.006 82	0.007 40
20	25	0.004 87	0.003 85
20	30	0.003 86	0.003 15
20	35	0.002 86	0.002 38
20	40	0.002 79	0.002 29

Table 2: Smoothed Good Toulmin and sequential Good-Turing evaluated on a Zipf(3)-distribution

n	m	Turing loss	Toulmin loss
10	20	0.039 33	0.085 96
10	15	0.024 86	0.050 74
10	11	0.017 05	0.014 47
10	25	0.029 50	0.077 70
10	30	0.044 13	0.119 62
20	25	0.031 90	0.025 89
20	30	0.029 61	0.016 13
20	35	0.024 64	0.027 14
20	40	0.031 01	0.041 16

Table 3: Smoothed Good Toulmin and Sequential Good-Turing evaluated on the Great Gatsby.

n	m	Turing loss	Toulmin loss
10	20	0.026 52	0.579 11
10	15	0.027 53	0.482 57
10	11	0.022 64	0.231 18
10	25	0.028 37	0.594 34
10	30	0.022 07	0.592 11
20	25	0.017 09	0.186 45
20	30	0.030 03	0.300 16
20	35	0.021 89	0.337 03
20	40	0.014 17	0.375 87

Table 4: Smoothed Good Toulmin and sequential Good-Turing evaluated on The Picture of Dorian Gray.

n	m	Turing loss	Toulmin loss
60	20	0.004 01	0.003 76
60	30	0.006 81	0.005 21
60	40	0.011 21	0.003 33
60	50	0.007 75	0.002 82
30	10	0.006 56	0.007 37
30	15	0.008 68	0.007 65
30	20	0.010 90	0.010 26
30	25	0.010 85	0.009 81

Table 5: Good Toulmin and sequential Good-Turing on data generated from a Zipf(2) distribution.

n	m	Turing loss	Toulmin loss
60	20	0.001 08	0.001 09
60	30	0.000 61	0.001 10
60	40	0.000 70	0.000 67
60	50	0.000 71	0.001 19
30	10	0.001 52	0.001 96
30	15	0.001 04	0.000 71
30	20	0.000 83	0.001 95
30	25	0.001 27	0.001 62

Table 6: Good Toulmin and sequential Good-Turing evaluated on a Zipf(3)-distribution

n	m	Turing loss	Toulmin loss
60	20	0.014 57	0.008 35
60	30	0.007 06	0.006 93
60	40	0.013 10	0.008 97
60	50	0.017 29	0.011 94
30	10	0.031 07	0.031 53
30	15	0.023 54	0.021 73
30	20	0.019 72	0.021 04
30	25	0.023 68	0.020 56

Table 7: Good Toulmin and sequential Good-Turing evaluated on the Great Gatsby.

n	m	Turing loss	Toulmin loss
30	10	0.014 23	0.006 61
30	15	0.007 30	0.007 09
30	20	0.024 10	0.022 95
30	25	0.018 91	0.019 15
60	20	0.012 44	0.007 49
60	30	0.012 41	0.002 55
60	40	0.008 60	0.007 01
60	50	0.007 20	0.006 31

Table 8: Good Toulmin and sequential Good-Turing evaluated on The Picture of Dorian Gray.

n	m	Turing loss	Toulmin loss
90	10	0.005 46	0.004 29
80	20	0.001 86	0.002 44
70	30	0.003 72	0.003 78
60	40	0.003 33	0.003 83
50	50	0.002 40	0.004 67

Table 9: Good Toulmin and sequential Good-Turing evaluated on Zipf(2) distributed data. (without using Monte-Carlo)

n	m	Turing loss	Toulmin loss
40	60	0.006 75	0.003 88
30	70	0.007 84	0.004 97
20	80	0.007 74	0.006 92
10	90	0.007 58	0.010 47

Table 10: Smoothed Good Toulmin and sequential Good-Turing evaluated on Zipf(2) distributed data. (without using Monte-Carlo)

n	m	Turing loss	Toulmin loss
40	60	0.000 82	0.000 67
30	70	0.000 99	0.000 60
20	80	0.001 22	0.001 11
10	90	0.001 18	0.001 48

Table 11: Smoothed Good Toulmin and sequential Good-Turing evaluated on Zipf(3) distributed data. (without using Monte-Carlo)

n	m	Turing loss	Toulmin loss
90	10	0.001 17	0.001 38
80	20	0.002 17	0.002 31
70	30	0.000 78	0.001 17
60	40	0.000 50	0.000 71
50	50	0.000 74	0.001 65

Table 12: Good Toulmin and sequential Good-Turing evaluated on Zipf(3) distributed data. (without using Monte-Carlo)

Master's Theses in Mathematical Sciences 2022:E54
ISSN 1404-6342

LUNFMS-3113-2022

Mathematical Statistics
Centre for Mathematical Sciences
Lund University
Box 118, SE-221 00 Lund, Sweden

http://www.maths.lu.se/