The Extended Maximum Likelihood Estimation for Monotone Probability Mass Function with Application using Forensic Data

Tiancheng Ma

Bachelor's thesis
2022:K2

Lund University

Faculty of Science
Centre for Mathematical Sciences
Mathematical Statistics

Centrum Scientiarum Mathematicarum

Acknowledgement

I would like to give my most sincere gratitude to supervisor Dragi Anevski for giving me the opportunity to work with him and for his supports and guidance through this work. To be able to complete this work and having contact with him gave me a huge amount of inspiration, motivation and ambition towards my studies in mathematical statistics.

I also would like to greatly thank Professor Marjan Sjerps of the Dutch Forensic Institute for providing forensic experiment data, which played an important role for this project.

Contents

Abstract

This paper presents solutions to the modelling of frequency data of species labels. but the data is incomplete in the sense that some rarely-occurring species labels give zero observed frequency because those species are rare. The data can be modelled by a monotone probability function with parameters to be estimated, and yet, due to the order constraints and the incomplete data, using conventional parameter estimation methods will cause trouble. Therefore, we study a previously introduced method to resolve the issue.

We begin with a brief tour through the attempts of parameter estimation, starting with the estimators which leads to problematic situations when using them. After that, we will study the improved estimation introduced in [1], which resolves the problem.

The improved estimation method seems to perform well when dealing with the problem, and it may be especially useful in the fields of forensic science, zoology, medical science, business analytic and even several fields of machine learning, especially pattern recognition.

1 Introduction

In this project we will discuss the methods of modelling a monotone species labels data as a probability mass function with constrains, using data with missing observations due to rarity of some species labels.

The species labels data comes from [2], which is a case study of gene species investigation collected from 2085 Dutch males. It is given a data table of gene species labels frequencies, using various types of gene detection technologies/methods.

The interest is to estimate the model's parameter through inference techniques introduced in [1]. We will begin by showing some naive estimation methods such as empirical distribution estimator and non-parametric maximum likelihood estimator in Section 2, with their limitations of giving problematic results. Next, we will introduce the extended and improved estimation: the pattern/profile maximum likelihood (PML) estimation, and with it, the sieved pattern/profile maximum likelihood (sPML) estimator which is a modified version of PML with optimized computation efficiency, they will be presented in Section 3, together with some properties. The experimental result with the gene species data will be presented in Section 4. Finally, the conclusions, will be given in Section 5.

2 Statement of the problem

Let us begin with the following example scenarios:

1. A zoologist collected data from a field study consisting of the frequency table for all animal species in a region. The zoologist has a list of all the known animal species in the region, but within those animal species, some animal species are too rare to be seen and resulted in zero observations for those.

2. Clinical data consists of frequencies of different symptoms occurring when the patients is infected by a type of bacteria/virus. According to biological theories, patients should suffer a range of symptoms, but within those symptoms there exists some very rare symptoms that do not occur so often on patients, therefore not observed.

3. We have the word statistics of one of the Shakespeare's texts, one wonders how many rare words does Shakespeare know?

The above scenarios could have their data modelled as a multinomial distribution. Let us define the parameter space for the multinomial distribution as:

$$\Theta = \{(\theta_1, \theta_2, ..., \theta_n, \theta_{n+1}) : \theta \in [0,1]^n, \theta_1 \geq ... \geq \theta_n, \sum_{i=1}^{n+1} \theta_i = 1\}$$

Having established the distribution and parameter space, let us turn to the parameter estimation methods. One can attempt first by using the empirical distribution estimator, which will be refereed as "empirical estimation/estimator" throughout this paper. Define an ordered set of data $T = \{t_1, t_2, ..., t_{n+1}\}$ where t_1 is the frequency of a specie that is the most common and t_{n+1} is the frequency of a specie that is the rarest, furthermore, given the theoretical order for the species labels likelihood as known. The empirical estimator is defined as following:

$$\hat{\theta}_k = \frac{t_k}{n}, \; k = 1, 2, ... n+1, \tag{1}$$

where n is the sample size. i.e. total numbers of observations. This estimation is straightforward, it works well for complete data entries and KNOWN theoretical likelihood order. However, for incompletely-surveyed data sets with zero data entries, for some rare species labels, this estimator give 0 for those rare species labels. In such case, the empirical estimator is not the most suitable.

Now, let us consider the case if the theoretical order of species labels' likelihood is unknown, one could approach the problem using the non-Parametric maximum likelihood estimator (NPMLE). Define χ as a bijective permutation which mixes the species labels. Define the data set $N = (N_1, N_2, N_3, ...)$ with unknown theoretical order and sample size n, then we will have the following discrete probability measure:

$$P^{(n,\theta)}(A) = \sum_{(N_1,N_2,N_3,...)\in A} \binom{n}{N_1 N_2 ...} \sum_{\chi} \prod_{i} \theta^{N_i}_{\chi(i)}, \tag{2}$$

with $\theta^{N_i}_{\chi(i)}$ denotes the species labels likelihood corresponding to N_i, and also re-arranged by the function χ. The NPMLE for (2) will be defined as following:

$$L(\hat{\theta}) = \arg \max_{\theta_1 \geq \theta_2 \geq \theta_3 \geq ...; \sum \theta_i = 1} \sum_{\lambda} \prod_{i} \theta^{N_i}_{\chi(i)} \tag{3}$$

At this point, the problem is that this estimator does not always exists. To show this, let us assume we have a very small dataset, $N = (1, 1)$, with sample size $n = 2$, we have our $L(\hat{\theta})$ as

$$\sum_{\chi} \prod_{i} \theta^{N_i}_{\chi(i)} = 2(\theta_1\theta_2 + \theta_1\theta_3 + ... + \theta_2\theta_3 + \theta_3\theta_4 + ...)$$

Now, let us consider the half likelihood:

$$\theta_1\theta_2 + \theta_1(1 - (\theta_1 + \theta_2)) + \theta_2(1 - (\theta_1 + \theta_2)) + \mathcal{O}$$

where $\mathcal{O}$ includes all terms with indices above 3. If we differentiate with respect to θ_1, we will see that it yields maximum if and only if $1 - 2\theta_1 - \theta_2 = 0$.

In fact, because the likelihood is symmetric in parameters, we will have $1 - 2\theta_i - \theta_j = 0$, $i \neq j$, for all positive integers i and j. hence, the maximum is attained at $\theta_i = \theta_j$.

Now suppose that the cardinality of species is nearly infinite. i.e. there are $\aleph < \infty$ numbers of species, then clearly the solution to this will be $\hat{\theta} = (\frac{1}{\aleph}, \frac{1}{\aleph}, ...)$, which cannot sum up to 1 and the order assumption of parameters will not satisfy. This example is taken from [1]. of page 30-31.

In the next section, we will introduce an improved and extended version of NPMLE: the Profile/Pattern Likelihood Estimator, which is proven to exist and to be consistent by [1].

3 Extended model: the pattern likelihood estimator

As we have showed the inference problem in Section 2, does not necessary have a solution. In this section, a new estimator will be introduced to overcome the inference problem. The new estimator is an extended estimator of NPMLE. and it has some interesting properties.

3.1 Formulation of the pattern maximum likelihood estimator

In order to address the inference problem, let us introduce the $\theta_0 = 1 - \sum_1^n \theta_i$ as the likelihoods collection of the "blob-species", which are the species labels that are very likely to generate zero observation. Thus, with the new component, the Pattern Maximum Likelihood (PML) Estimator is defined as follows:

$$\arg \max_{\theta:\theta_1 \geq \theta_2 \geq \theta_3 \geq \ldots; \sum_{\alpha=1}^{\infty} \theta_\alpha \leq 1} \sum_{\chi} \frac{n!}{N_0! \prod_{i \geq} N_i!} \theta_0^{N_0} \prod_{\alpha=1}^{\infty} \theta_\alpha^{N_{\chi}^{-1}(\alpha)}. \tag{4}$$

Another version of the PML is the Sieved Profile Maximum Likelihood Estimator (sPML) which is a modification to the PML dedicated to optimize computation, derived from [1].

Consider the true parameter set p. let us define $\tilde{q}$ as the truncated vector of likelihood with indices $\{1, ..., k\}$, where k is a truncation level that separate blob and non-blob species, and $\tilde{q}_1 \geq \tilde{q}_2 \geq \tilde{q}_3 \geq \ldots \geq \tilde{q}_k$. $q_0 = 1 - \sum_{\alpha=1}^{k} \tilde{q}_\alpha$.

Now, suppose the ordered observed data set $N = (N_1, N_2, N_3, ...)$ with sample size n, where $N_1 \geq N_2 \geq N_3 \geq ....$ The data has an underlying multinomial distribution $X = (X_1, X_2, X_3, ...)$ with parameters $(q_0, \tilde{q})$. Define X_+ as $\sum_{\alpha=1}^{k} X_\alpha$ accordingly to the truncated level. Then denote the truncated sample set N_+ as $(N_1, N_2, ..., N_s)$. where s is the number of specie labels of non-blob-species, furthermore, $N_1 \geq N_2 \geq N_3 \geq ... \geq N_s$.

Next, append X_0 number of ones to the truncated sample set, so N_+ becomes $N = (N_1, N_2, ..., N_s, 1, 1, 1, ...)$. X_0 is the number of unobserved species labels.

Ultimately, using the truncated sample set N_+, the sieved species likelihood $(q_0, \tilde{q})$ and the same bijective permutation function χ, let us formulate the sieved likelihood as the following:

$$L(\tilde{q}) = \sum_{\chi} \frac{n!}{N_0! \prod_{i \geq 1} N_i!} q_0^{N_0} \prod_{\alpha=1}^{\infty} \tilde{q}_\alpha^{N_{\chi^{-1}(\alpha)}},$$

with $N_0 = n - \sum_{\alpha=1}^{k} N_{\chi^{-1}(\alpha)}$, where

$$\tilde{q} = \arg \max_{\tilde{q}: \tilde{q}_1 \geq \tilde{q}_2 \geq \tilde{q}_3 \geq ... \geq \tilde{q}_k; \sum_{\alpha=1}^{k} \tilde{q}_\alpha \leq 1} L(\tilde{q})$$

3.2 Existence and Consistency

The PML and the sPML have some interesting properties.

According to [1], the extended Maximum Likelihood estimator exists, which can be summarized by the following theorem:

Theorem 1. *(i) Under the topology of point-wise convergence, the parameter space Θ is compact.*
(ii) The functional $L : \Theta \mapsto \mathbb{R}_+$ defined by:

$$L(\theta) = \sum_{\chi} \frac{n!}{N_0! \prod_{i \geq} N_i!} \theta_0^{N_0} \prod_{\alpha=1}^{\infty} \theta_\alpha^{N_{\chi^{-1}(\alpha)}}$$

with $N_0 = n - \sum_{\alpha=1}^{\infty} N_{\lambda^{-1}(\alpha)}$, is a continuous functional.

Thus, the extended model, the pattern/profile maximum likelihood estimator defined in (4), exists.

The PML estimator is consistent in the form of almost-surly convergence in L1-norm, as the sample size gets larger, i.e.

$$||\hat{\theta}^{(n)} - \theta||_1 \to 0 \ \ as \ \ n \to \infty$$

where $\hat{\theta} = \hat{\theta}^{(n)}$ be the PML estimator. For any $\delta > 0$, the following weak consistency result of the PML estimator is obtained, according to [1]:

$$P^{n,\theta}(||\hat{\theta}^{(n)} - \theta||_1 > \delta) \leq \frac{1}{\sqrt{3n}} e^{-\pi\sqrt{n}(\sqrt{n}\frac{\epsilon^2}{2} - \sqrt{\frac{2}{3}})}(1 + o(1)), \tag{5}$$

as $n \to \infty$. Here, $\epsilon = \frac{\delta}{4r}$ and $r = (\delta, \theta)$ as a function of δ and θ, such that $\sum_{r+1}^{\infty} \theta_1 < \frac{\delta}{4}$.

As explained in [1], the sequence of maximum likelihood estimators $\hat{\theta}^{(n)}$ is strongly consistent in L1-norm, i.e.

$$\sum_{i=1}^{r} |\hat{\theta}_i^{(n)} - \theta_i| \xrightarrow{a.s.} 0.$$

This is a consequence of (5), by the characterization $X_n \xrightarrow{a.s.} 0 \iff \sum_{i=1}^{\infty} P(|X_n| > \delta) < \infty$, with $X_n = ||\hat{\theta}^{(n)} - \theta||_1$ since

$$\sum_{n=1}^{\infty} \frac{1}{\sqrt{3}} e^{-\pi\sqrt{n}(\sqrt{n}\frac{\epsilon^2}{2} - \sqrt{\frac{2}{3}})} < \infty$$

The above consistency results suggest that if the sample size increases, then the L1-error should go to zero.

In order to investigate the behaviour of consistency, and some additional properties, the simulation studies for each of them will be introduced, in the next subsection.

3.3 Simulation Study

The simulation studies are performed on R-studio using the Stochastic Approximation EM (SA-EM) algorithm for PML estimator developed in [1]. First, let us perform the data simulation. To do this, set a true parameter vector $\theta = (\theta_1, \theta_2, \theta_3, ...)$ consisting of positive, monotonously decreasing parameters that sums to 1. Then, use the true parameter vector to generate multinomial data $X_{sim}^{(n)} = (x_1, x_2, x_3, ...)$, where n is the sample size, and

$x_1, x_2, ...$ are data generated with indices corresponding to the true parameter vector elements. At last, use the sample to perform estimations and conduct various experiments.

In this section, additional PML's consistency properties and the behaviour of estimation error will be investigated. Then, the PML estimator will be compared with the Good-Turing estimator. Finally, an experiment with statistical bootstrapping will be carried out, as a way to generate confidence interval for the PML estimation.

3.3.1 Consistency of the PML estimator vs the empirical estimator

The first part of the simulation study is to study PML estimation's consistency with different true parameter vectors.

Consider a short true parameter vector of length 6:

$$\theta = (0.4, 0.2, 0.15, 0.1, 0.1, 0.05). \tag{6}$$

Then, let us loop the following process: First, setting the sample size n to 1000 and use the vector (6) to generate multinomial data, denoted as $X_{sim}^{(n)}$. Then, use $X_{sim}^{(n)}$ to perform PML estimation and the empirical estimation, denoted as $\hat{\theta}^{(n)}$. Finally, calculate the L1-error, i.e

$$d(\hat{\theta}^{(n)}, \theta) = \sum_{i=1}^{m} |\hat{\theta}_i^{(n)} - \theta_i|.$$

The sample size n of the simulation data-set is increased by 1000 when the process loop backs to the start. Let us perform the above process 1000 times, then the L1-error will be stored as a vector, called

$$\eta = (d_1(\hat{\theta}^{(n)}, \theta), d_2(\hat{\theta}^{(n)}, \theta), ..., d_t(\hat{\theta}^{(n)}, \theta)),$$

where t is the number of loops.

Once the error vector is computed, we illustrate the error as a scatter plot. Eventually, The diagram is generated and shown in Figure 1. Observe

from the plot, the L1-error of the empirical estimator and the PML estimator seem to decrease exponentially, as n increases, and that the L1-error for the PML estimator is generally higher.

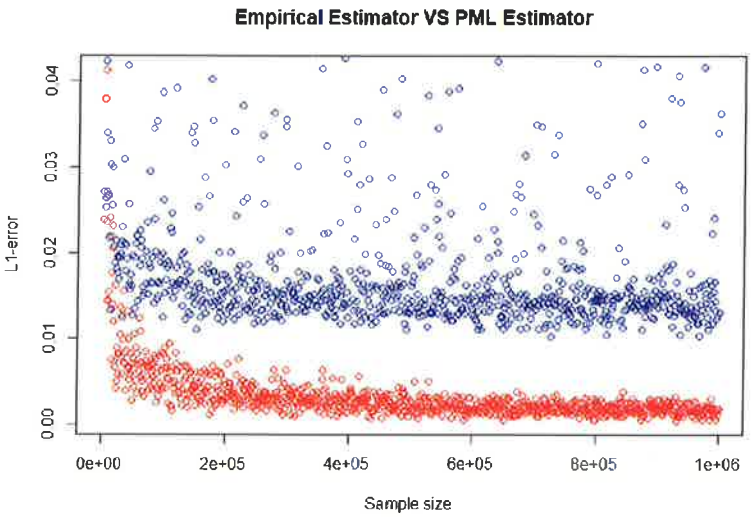


Figure 1: *L1-Error plot of PML (Blue) and the Empirical Estimation (Red)*

The result in Figure 1 shows that using PML to estimate small amount of species labels will be less accurate than using the Empirical estimator. In addition, it seems like the PML estimator has plenty of error jumps. In such case, the empirical estimator seems to perform better for small amount of specie labels.

Let us begin another experiment with a longer true parameter vector with 278 species labels:

$$\theta = (0.1, 0.1, 0.5, 0.5, 0.5, 0.008, 0.008, 0.008, 0.008, \theta_{10}, \theta_{11}, \theta_{12}, \theta_{13}), \quad (7)$$

where θ_{10} is a vector with twenty 0.005's, θ_{11} is a vector with sixty 0.004's, θ_{12} is a vector with eighty-nine 0.002's and θ_{13} is a vector with one-hundred

0.001's. The vector (7) will be used and go through the same procedure. Finally, the result is generated in Figure 2:

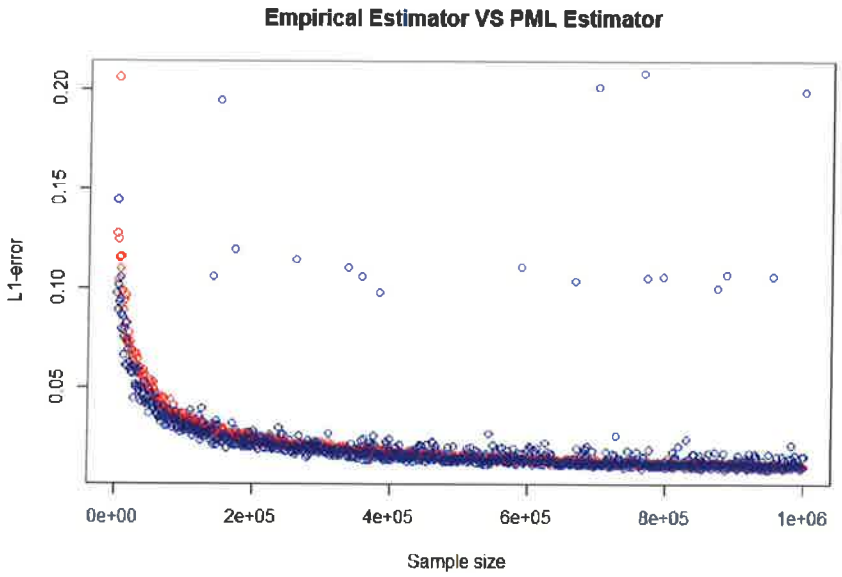


Figure 2: *Scatter plot of L1-error against sample size, of the PML estimator (blue scatters) and the Empirical Estimator (red scatters)*

A further observation from Figure 2 is that the performance of PML Estimation is nearly as good as the Empirical estimation, which should be compared to the result in Figure 1. However, the random erroneous jumps still exist, but thankfully they occur less often, compared the result in Figure 1.

To better observe the pattern, let us plot the result of Figure 2 using logged-scale, see Figure 3:

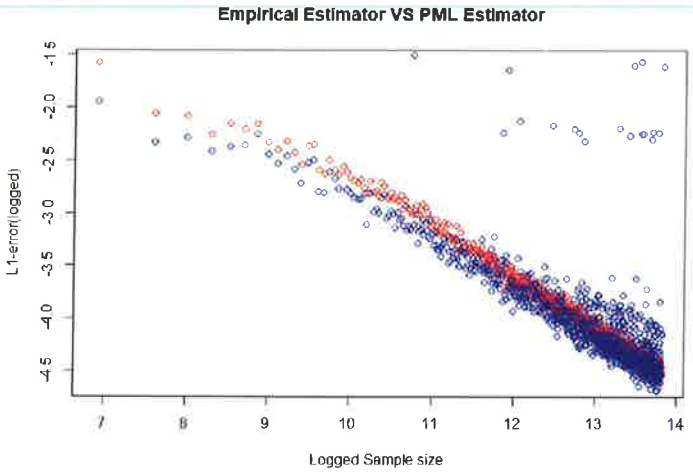


Figure 3: *X-Y Log-scaled scatter plot of L1-error against sample size, of the PML estimator (blue scatters) and the Empirical Estimator (red scatters)*

Observe from Figure 3, the L1-error tends to go to 0 as we increase the sample size, in the manner of asymptotic linear trend.

The theoretical assumption of determining the convergence rate α is that $n^{\alpha}||\hat{\theta}^{(n)} - \theta||_1 \xrightarrow{a.s.} C$, for some $0 < C < \infty$. Thus for large number of n, $n^{\alpha}||\hat{\theta}^{(n)} - \theta||_1 \simeq C$, with probability 1. Taking the logarithm on both sides we have $\log||\hat{\theta}^{(n)} - \theta||_1 \simeq \log C - \alpha \log n$. Therefore, the convergence rate can be determined by a plot of logged L1-error versus logged sample, which is essentially Figure 3. Thus, the slope should be α and this can be determined by using linear regression analysis.

The plot with the regression lines is illustrated in Figure 4. As the result, the slope is -0.436 for the PML estimation, which refers to the result in [1] says that the convergence rate is faster than estimation: $-\alpha = -(\frac{1}{4} + \epsilon)$ for any $\epsilon > 0$. Note that it cannot be faster than the parametric rate $n^{\frac{1}{2}}$

If linear regression is applied once again to determine the rate of convergence for the empirical estimator, then the slope obtained is -0.495 as the

result, which is quite close to its optimal rate $-\alpha = -\frac{1}{2}$. The conclusion from this study is that the convergence rate for the PML estimator is nearly as good as the one for the empirical estimator.

The regression line is added to the log-scaled scatter plot, see Figure 4 below:

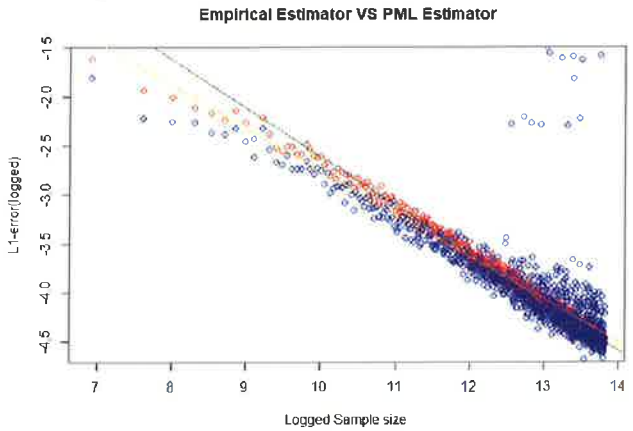


Figure 4: *X-Y Log-scaled L1-error plot against sample size, with Regression line. Yellow line represents the convergence rate of the PML estimation error, and the dark line represents the same thing for the Empirical Estimation error.*

3.3.2 Comparison with Good-Turing Estimation

This experiment will compare the ability of estimating blob-species labels likelihood (the species label likelihood that likes to give zero observation) between PML estimator and the Good-Turing estimator.

Definition. *Consider an ordered sample data*

$$X = (X_1, X_2, ..., X_{k-1}, X_k, X_{k+1}, ..., X_{n-1}, X_n, X_{n+1}, .., X_{s-1}, X_s)$$

where $X_1 \geq X_2 \geq ... \geq X_s$. *The* k, n, s *are arbitrary numbers and* $k < n < s$. *In addition,* $X_{k+1} = X_{k+2} = ... = X_n = 1$ *are singletons, and* $X_{n+1} =$

$X_{n+2} = ... = X_s = 0$. *The* ***Good-Turning Estimator*** *is defined as the sum of the singletons divided by the sample data, i.e:*

$$P_{Good-Turing} = \frac{\sum_{i=k+1}^{n} X_i}{\sum_{i=1}^{s} X_i}$$

Note that in the context of specie labels frequency estimation, the Good-Turing estimating the likelihood of observing rare species labels becomes the number of species labels seen exactly once divided by the total number of samples that has been observed.

Let us denote $\theta_0^{(n)}$ as the total species likelihood PML estimates of zero-observed species estimates when sample size equals n, and $P_{Good-Turing}^{(n)}$ represents the Good-Turing estimator, when sample size equals n. Next, define the sum of the species labels likelihood that gives zero observation. θ_0. The difference between θ_0 and $P_{Good-Turing}^{(n)}$ is:

$$\eta_{Good-Turing} = \{\theta_0 - P_{Good-Turing}^{(0,n)}, \theta_0 - P_{Good-Turing}^{(1,n)}, ..., \theta_0 - P_{Good-Turing}^{(k,n)}\} \quad (8)$$

One could also define the identically formulated difference between θ_0 and the PML estimates of blob-species $\theta_0^{(n)}$ as:

$$\eta_{PML} = \{\theta_0 - \theta_0^{(0,n)}, \theta_0 - \theta_0^{(1,n)}, ..., \theta_0 - \theta_0^{(k,n)}\} \quad (9)$$

The experiment will be performed using data generated by the long true parameter vector (7), set the repeat times $t = 300$ times, with fixed sample size $n = 2000$. The result of accuracy falloff ηs for both PML and Good-Turing estimator, are presented in Figure 5

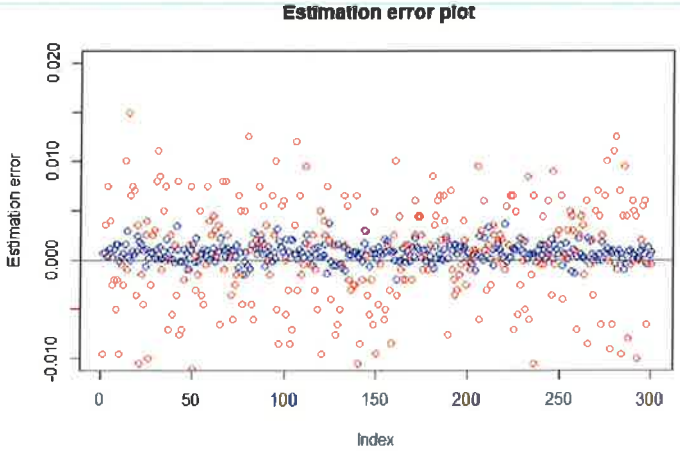


Figure 5: *The scatter plot of the estimation error between the sum of the true parameter vector's rare species and correspond to the PML Estimator (Blue Scatters) and the Good-Turing Estimator (Red Scatter)*

From Figure 5, the PML estimator behaves consistent when estimating the rare species, in contrast, the Good-Turning estimator has high variance.

3.3.3 Distribution of the Estimation Error

The behaviour of L1-error $|\hat{\theta}_{PML} - \theta|$, in terms of its distribution, will be investigated in this subsection.

To begin with, let us generate the data with the true parameter vector (7), with a fixed sample size n. Then compute the L1-error of the PML estimator. Repeat this process many times, to generate a vector of L1-error for each repetition. let us call it η.

The density of η can be estimated using kernel density estimator (KDE), it was originally worked out by Emanuel Parzen (1962) and Murray Rosenblatt

(1956). The expression for the KDE is

$$\hat{f}_n = \frac{1}{n}\sum_{1}^{n} K_h(x - x_i).$$

given identically independent data $(x_1, x_2, ..., x_i)$ and K_h is the kernel functional, such as uniform, triangular, normal kernels, given bandwidth h. In this simulation study, we will use the default kernel density settings in R-studio with automatic bandwidth and normal kernel. Finally, the last step is to use the Kolmogorov-Smirnov test to validate with a few data generated by the hypothesis distribution of parametric distributions. The test consist of the following hypothesis:

H_0: The distribution of η behaves like our hypothesis distribution

H_1: The distribution of η does NOT behave like our hypothesis distribution

If the p-value is generally greater than 0.05, we will not reject the null hypothesis.

Now, let us use the parameter vector (7) to generate data with a fixed sample size to $n = 40000$, compute the vector η. Finally, generate the following plot of the KDE estimation of η:

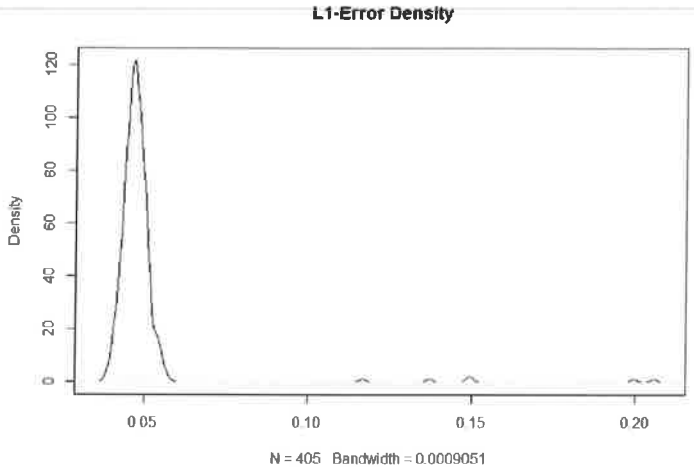


Figure 6: *The L1-Error η Density plot generated from the Kernel Density Estimation*

Looking at the plot, we noticed that majority of the data is centered around 0.05, and there exists some other small entities scatters beyond $x = 0.1$. Let us make Gaussian distribution as a hypothesis.

Construct a Gaussian random variable with expectation 0.04712 and standard deviation $\sigma = 0.0032$, then perform the Kolmogorov-Smirnov Test, the p-value seems generally greater than 0.05 (on average larger than 0.2, sometimes reaching 0.6), which means that the null hypothesis cannot be rejected, therefore, the L1-error seems to behave similar to the Gaussian Random Variable.

3.3.4 An attempt to generate confidence interval using statistical bootstrapping

To prepare for the next subsection on the data analysis. Let us try to generate a 95% Confidence Interval (CI) for the PML estimation, based on the statistic

$(\hat{\theta}_n - \theta)$ in terms of its distribution, given the simulation data.

Usually, to obtain a confidence interval for an estimate, one needs to gather multiple dataset by repeating many data gathering process, which requires large efforts and costs many resources. To overcome the problem, we will use the bootstrapping method introduced by B. Efron in 1979, in [4].

The general idea of bootstrapping, in this paper, is to first use the gathered data $x_1, x_2, ..., x_n$ to estimate the "reference" PML estimator $\hat{\theta}_n$ using the data $x_1, x_2, ..., x_n$. Then, generate bootstrapping partitions with replacement, call them bootstrapping samples $x_1^*, ..., x_n^*$ based the collected data $x_1, x_2, ... x_n$. Afterwards, make use of bootstrapping samples to compute the bootstrapped PML estimators $\hat{\theta}_n^*$. Finally, compute $\hat{\theta}_n^* - \hat{\theta}_n$, and let us call that statistic as $\eta = \hat{\theta}_n^* - \hat{\theta}_n$. This is one complete step of bootstrapping estimation.

We repeat the bootstrapping estimation process k-many times by the following manner:

1. Generate the simulation data $(x_1, x_2, ..., x_n)$ using the true parameter vector (7). Estimate the "reference" PML estimator $\hat{\theta}_n$, based on $(x_1, x_2, ..., x_n)$.

2. Do the bootstrap data sampling k-many times, to generate k number of bootstrap sample vectors:

   $$x^{*(1)} = (x_1^{*(1)}, ..., x_n^{*(1)}), x^{*(2)} = (x_1^{*(2)}, ..., x_n^{*(2)}), \ ..., \ x^{*(k)} = (x_1^{*(k)}, ..., x_n^{*(k)}).$$

   This simulates the repeated data gathering process that gives random species labels distribution

3. Using each bootstrap partition vectors, calculate the bootstrapping PML estimators to obtain a vector of bootstrap estimates, which is

   $$\hat{\theta}_n^* = (\hat{\theta}_n^{*(1)}, \hat{\theta}_n^{*(2)}, ..., \hat{\theta}_n^{*(k)}).$$

   Where $\hat{\theta}_n^{*(i)}, 1 < i < k$ denotes the PML estimator of the bootstrapping partition vectors with corresponding indices from 1 to k.

4. Make use of the "reference" PML estimation $\hat{\theta}_n$ and each row vector of the bootstrap PML estimator $\hat{\theta}_n^{*(i)}$ to calculate the bootstrapping statistic $\hat{\theta}_n^{*(i)} - \hat{\theta}_n$ for integer $1 < i < k$. Then generate a bootstrapping statistic matrix. Note that each column corresponds to a specie index, and each row corresponds to a bootstrap index. define $k \times \alpha$ bootstrapping error matrix in the following, where k stands for number of bootstrapping partition vectors and α number of species labels:

$$\hat{\eta}_n^* = \begin{pmatrix} \hat{\theta}_n^{*(1)} - \hat{\theta}_n \\ \hat{\theta}_n^{*(2)} - \hat{\theta}_n \\ \vdots \\ \hat{\theta}_n^{*(k)} - \hat{\theta}_n \end{pmatrix}$$

Each column of $\hat{\eta}$ is a column vector with length k, consists of likelihood estimation error of each species labels, after bootstrapping k-times.

5. Take each column of $\hat{\eta}_n^*$ to generate the bootstrap likelihood estimation error distribution of the "reference" PML estimator $\hat{\theta}_{(n)}$, so we can use it to generate a confidence interval for $\hat{\theta}_{(n)}$.

Each column of $\hat{\eta}_n^*$ can be used to determine an underlying distribution $\mathbf{Z}$. Based on $\mathbf{Z}$, the 95% confidence interval is formulated as:

$$\mathbf{I}_\theta = (\hat{\theta}_n + \mathbf{Z}_{0.975},\ \hat{\theta}_n - \mathbf{Z}_{0.025}),$$

In other words, once we have $\mathbf{Z}$ determined from the bootstrapping process, we will then obtain the probability

$$0.95 = Pr(\mathbf{Z}_{0.975} \leq (\hat{\theta}_n^* - \hat{\theta}_n) \leq \mathbf{Z}_{0.025}).$$

Now, let us use true parameter vector (7) to generate data with a fixed sample size $n = 4000$, we make a "reference" estimate $\hat{\theta}$ and perform the bootstrapping loops. Set k=600, so there will be 600 bootstrapping partitions, then compute the confidence interval and apply the CI to the "reference" PML estimation. Finally, generate the following plot in Figure 8:

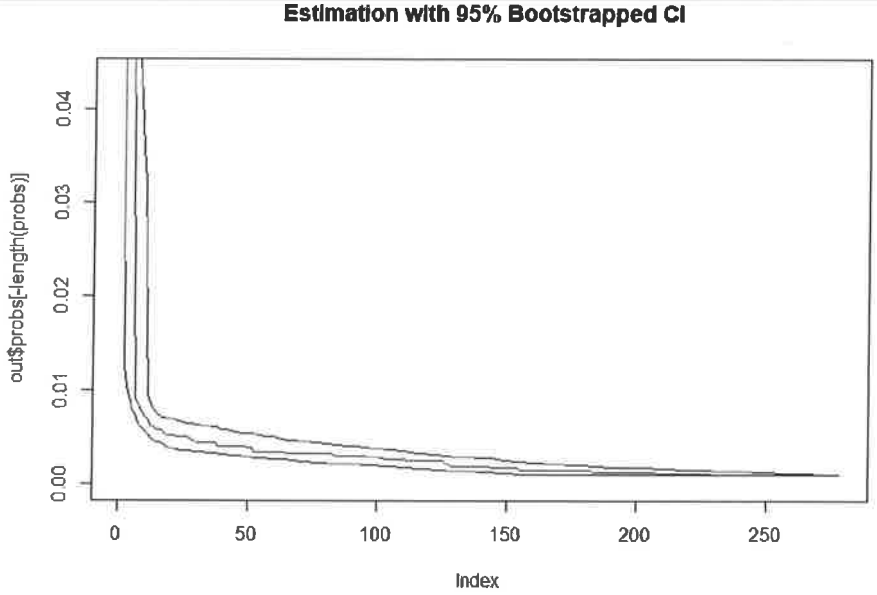


Figure 7: The PML estimation (Blue line) with confidence interval band (black lines)

Observed from Figure 8, the error of the estimation is fairly large for the common species labels, which are the ones with lower specie indices, and that become smaller very quickly as the species labels increases.

3.4 Summary

In this section the Pattern Maximum Likelihood estimator has been introduced, together with the Sieved Pattern Maximum Likelihood estimator to overcome the inference problem. Both PML and sPML have some interesting properties such as existence and strong consistency, which are further studied by a simulation study.

The PML estimator, Good-Turing Estimator and the Empirical estimator are compared in the simulation study. As the result, the empirical estimator seems to perform well overall, but it cannot estimate the rare species labels likelihoods which give zero observations. The PML estimator, in contrast, cannot estimate the likelihood of the common species labels accurately, but it does very well when estimating the likelihood of rare species labels. Another result is that the PML estimator performs very well with more species labels and sample size, which partially verifies the result in [1]. And yet, the PML estimator does not perform well when the species labels are few. The final results of this section is the density of estimation error behaves similar to Gaussian distribution.

Further more, the confidence interval for PML estimates can be determined through statistical bootstrapping. This technique will be used in the actual data modelling in the next section.

4 The Data and the Experiment Result

Let us begin our experiment with the attempt by Orlitsky et al. Setting the vector $\mathbf{n} = (1, 2, 3, 4, 5, 6, 7)$ representing the number of times a specie label is observed, and the vector $\mathbf{r} = (123, 138, 86, 51, 17, 2, 6)$ represents number of species labels observed n_i-many times. The following plot is generated.

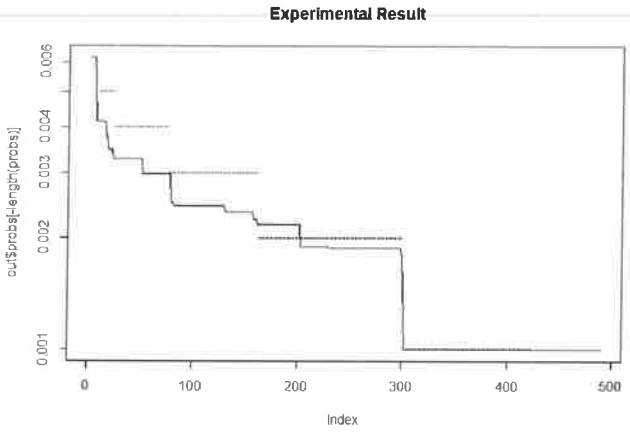


Figure 8: Experiment result of the attempt by Orlitsky et al.

The above plot represents the monotone-decreasing probability for each species labels on the x-axis "Species". Note that the dotted lines on the graph represents the estimation using the empirical estimator. The empirical estimator cannot estimate the blob-species while the PML estimator can.

Let us turn the focus on the genetic sample data from [2], with the genetic samples collected from 2085 Dutch man. The gene species data collected using different analyzing/detection methods is presented in Table 4 in [2].

To better understand the formulation of the data in Figure 10, we use "PPY" as an example. The data under "PPY" detects 938 gene species once, 145 gene species twice, 60 gene species three times, 24 gene species four times, and so on... They make total number of gene species is 1217, says on the bottom row of the table. Before we perform the parameter estimation, define vector **n** as the number of times a gene specie is observed, represents the very first column in the data table. and **r**, the number of gene species observed n_i-many times, represents the column under the name of a gene detection method. Example: The "Min YHRD" data. set

$$\mathbf{n} = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 17, 18, 19, 22, 23, 24, 25, 30, 53, 63, 84, 99, 107),$$

Uniqueness of Y-STR haplotypes in 2085 Dutch samples

Uniqueness: # times a haplotype was observed	# occurrences in 2085 samples					
	Min YHRD[a]	PPY[b]	Yfiler[b]	PPY23[b]	RMY 1+2[b]	PPY23+RMY 1+2
1	608	938	1650	1929	2052	2065
2	110	145	130	70	15	10
3	46	60	30	4	1	
4	28	24	7	1		
5	16	13	5			
6	16	11	2			
7	6	4	1			
8	5	4				
9	2	3				
10	4	1				
11	2					
12	2	1				
13	1	3	1			
14	1					
15		1				
17	3	2				
18	1	1				
19	1					
22	1					
23	1					
24	1	1				
25	1					
28		1				
30	1					
31		1				
43		1				
52		1				
53	1					
57		1				
63	1					
84	1					
99	1					
107	1					
# different haplotypes	862	1217	1826	2004	2068	2075

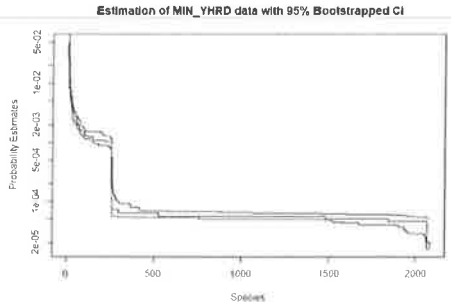
[a] The minimum YHRD marker set consists of DYS19, DYS385a, DYS385b, DYS389I, DYS389II, DYS390, DYS391, DYS392, DYS393.
[b] See Table 1 for the marker units present in these Y-STR sets.

Figure 9: The table of the haploid species (haplotype) data, collected from 2085 Dutch males. Source: Antoinette A. Westen et al

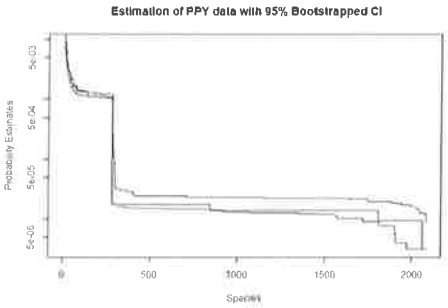
and

$$\mathbf{r} = (608, 110, 46, 28, 16, 16, 6, 5, 2, 4, 2, 2, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1).$$

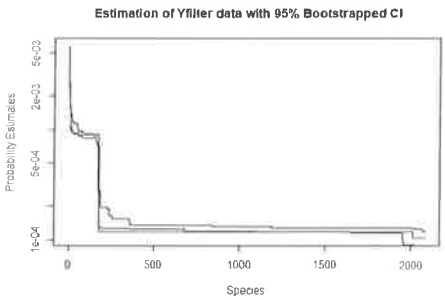
We have the following results after the implementation for all the data in Figure 10 and 11, except the column "PPY23 + RMY 1+2":



(a) Min YHRD Model

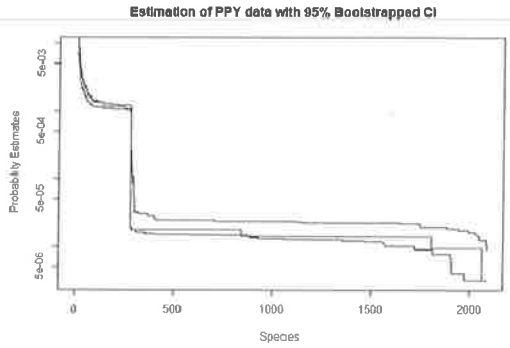


(b) PPY Model

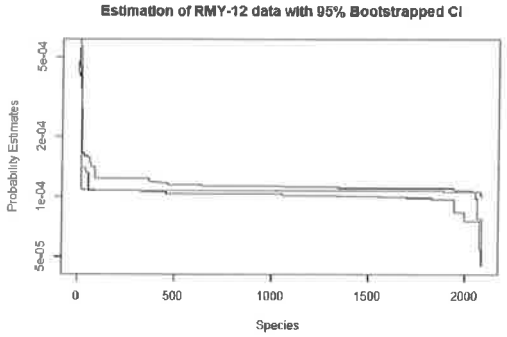


(c) Y-Filter Model

Figure 10: The estimation results, with the estimates in blue lines and confidence band in black lines



(a) PPY-23 Model



(b) RMY 1+2 Model

Figure 11: The estimation results, with the estimates in blue lines and confidence band in black lines (Continued)

Judging from the data estimation, we successfully modelled monotone decreasing function through the PML estimation. and the confidence interval via bootstrapping worked mostly fine.

However, the confidence interval for the PPY-23 and Yfilter data model is very faulty, this is unfortunately caused by the insufficient bootstrapping iterations. because of c computation power limitation, which often causes the PML estimatior algorithm to fail. Another problem was the modelling

data for "PPY23 + RMY (1+2)" failed, because nearly all the gene species are observed exactly once, which makes the likelihood fairly even: There are 2065 gene species has observed once and 10 gene species has observed twice. This will break the PML estimation algorithm.

5 Conclusion and Discussion

Throughout this study, the solution of the inference problem of modelling the species likelihood with incomplete data has introduced. The conventional methods of solving such inference problem, such empirical distribution estimator and the NPMLE, have presented together with their problems and limitations. After that, the PML estimator and its properties have introduced, which is then followed by a simulation study to further investigate the PML's behaviour.

The first result concluded from the simulation study is that the PML is consistent in L1-norm. However, the PML estimator is not the most optimal since it cannot estimate the likelihood of common species labels accurately. The second result is that the estimation error of PML estmator behaves similar to normal distribution. The final result is that the success of applying statistical bootstrapping to generate a confidence interval on PML estimator.

The gene data is modelled as a multinomial distribution, using the PML, and with bootstrapped confidence interval. It seems mostly successful, but due to limitation of computation and my lack of knowledge in optimizing the code and Rstudio environment, resulted in some of the confidence interval were not properly generated, and the data for "PPY23 + RMY (1+2)" cannot be modelled since the data structure is not suitable for the inference problem: It does not have a strong order since nearly all the gene species were observed exactly once.

The PML estimator has shown its potential in Forensic Science, from what this project studied on. It could also be useful in other fields, such as machine / deep learning, business analytic, medical science, which require a solution to the similar inference problem stated at the start of this paper.

References

[1] Anevski D, Gill R, Zohren S., (2017).

Estimating a Probability Mass Function with Unknown Labels

The Annals of Statistics vol.45, no.6, ISSN: 0090-5364

http://dx.doi.org/10.1214/17-AOS1542

[2] Antoinette A. Westen. et al., (2014).

Analysis of 36 Y-STR marker units including a concordance study among 2085 Dutch males

Forensic Science International: Genetics vol.14, no.1, ISSN: 1872-4973

https://www.sciencedirect.com/science/article/pii/S1872497314002282

[3] P Greselin., (2019).

Maximum likelihood estimation of a monotone probability mass function with unknown labels

LUP Student Paper id: 8984708

https://lup.lub.lu.se/student-papers/search/publication/8984708

[4] B Efron., (1979)

Bootstrap Methods: Another Look at the Jackknife

The Annals of Statistics Vol. 7 No.1, 1-26

Bachelor's Theses in Mathematical Sciences 2022:K2
ISSN 1654-6229

LUNFMS-4063-2022

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

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