scientific reports

OPEN

Bias in Zipf's law estimators

Charlie Pilgrim[1]✉ & Thomas T Hills[2,3]

The prevailing maximum likelihood estimators for inferring power law models from rank-frequency data are biased. The source of this bias is an inappropriate likelihood function. The correct likelihood function is derived and shown to be computationally intractable. A more computationally efficient method of approximate Bayesian computation (ABC) is explored. This method is shown to have less bias for data generated from idealised rank-frequency Zipfian distributions. However, the existing estimators and the ABC estimator described here assume that words are drawn from a simple probability distribution, while language is a much more complex process. We show that this false assumption leads to continued biases when applying any of these methods to natural language to estimate Zipf exponents. We recommend that researchers be aware of the bias when investigating power laws in rank-frequency data.

If we take a book and rank each word based on how many times it appears, we will find that the number of occurrences of each word is approximately inversely proportional to its rank[1]. The second most frequent word will appear approximately $\frac{1}{2}$ as often as the most frequent word, the third around $\frac{1}{3}$ as frequently. This describes a power law relationship between the frequency of a word, n, and the word's rank in terms of its frequency, r_e, with exponent $\gamma \approx 1$[2].

$$n(r_e) \propto r_e^{-\gamma} \tag{1}$$

This is known as Zipf's law and is consistent, in a general sense, across human communication[3,4]. We do not have a satisfactory reason why this is[2] and the exponent, γ, is not always 1 but varies between different speakers[3] and texts[3,5]. Sound analytical tools are needed to investigate these research areas.

Equation (1) describes an observed empirical relationship. It is tempting to assume that this is equivalent to a probability distribution for words (an early example is of Shannon using Zipf's law to estimate the entropy of English[6]). Indeed, Zipf's law is often expressed as a relationship between a word's probability of occurrence[7,8] and the word's rank in the probability distribution, r_p.

$$p(r_p) \propto {r_p}^{-\lambda} \tag{2}$$

The conflation of Eqs. (1) and (2) causes the prevailing maximum likelihood estimators to miscalculate λ in Eq. (2) with a positive bias[2,9,10] (Fig. 1). This bias applies specifically to rank-frequency distributions, where the ranks of events are not known a priori and instead are extracted from the frequency distribution, as is the case with word frequencies. The root of the bias is that the existing estimators make the assumption that the observed empirical frequency rankings of data [r_e in Eq. (1)] are equivalent to rankings in an underlying probability distribution [r_p in Eq. (2)][2]. The nth most frequent word is assumed to be the nth most likely word, which is not necessarily the case[2]. This is often overlooked in the literature[2].

In the 2000s there were a series of papers[8,11,12] describing a method of maximum likelihood estimation that gave more accurate (lower bias) estimates for power law exponents than graphical methods[8]. The most influential of these is Clauset et al.'s paper[8]. The estimators had been derived and presented before[11] (as early as 1952 in the discrete case[13]) but Clauset et al.'s paper popularised the idea and provided a clear methodology including techniques to perform goodness of fit tests[8]. In all of these papers, the derivation of the likelihood function assumes that there is some a priori ordering on an independent variable. This works very well for power laws with some natural way to order events, such as the size vs frequency of earthquakes[8]. However, it does not work so well with rank-frequency distributions, where the rank is extracted empirically from the frequency distribution, so that the empirical rank and frequency are correlated variables[2], both dependent on the same underlying mechanism. This difference was not addressed by Clauset et al., who include examples of applying their estimator to Zipf's law in language[8]. The same data can look very different depending on whether we know it's true rank or not, as shown in Fig. 2.

[1]Mathematics for Real-World Systems Centre for Doctoral Training, The University of Warwick, Coventry CV4 7AL, UK. [2]Department of Psychology, The University of Warwick, Coventry CV4 7AL, UK. [3]The Alan Turing Institute, British Library, 96 Euston Road, London NW1 2DB, UK. ✉email: charlie.pilgrim@warwick.ac.uk

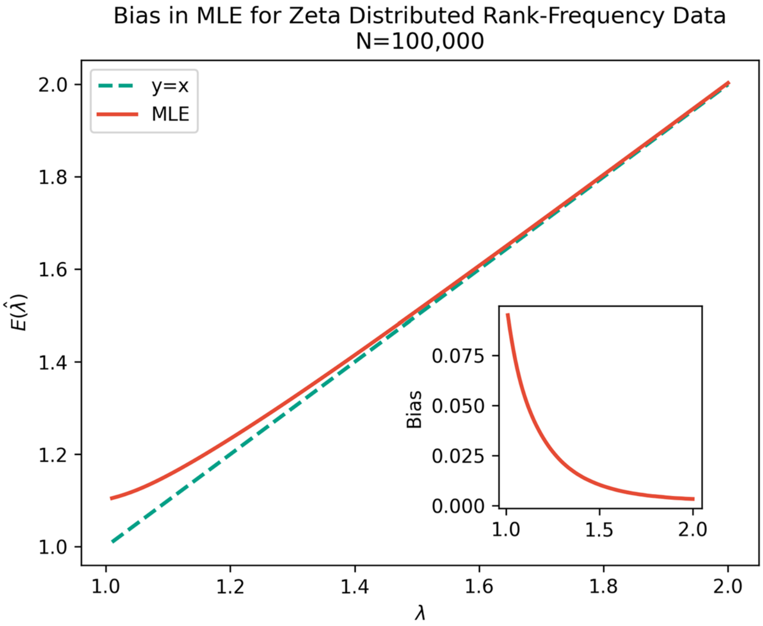

Figure 1. Bias in maximum likelihood estimation for rank-frequency data. 100 values of λ between 1 and 2 were investigated. For each λ, samples with $N = 100,000$ were generated from an unbounded power law distribution and Clauset et al.'s estimator was applied to the empirical rank-frequency distribution. This was repeated 100 times and results averaged. There is a clear and strong positive bias for $\lambda \lessapprox 1.5$.

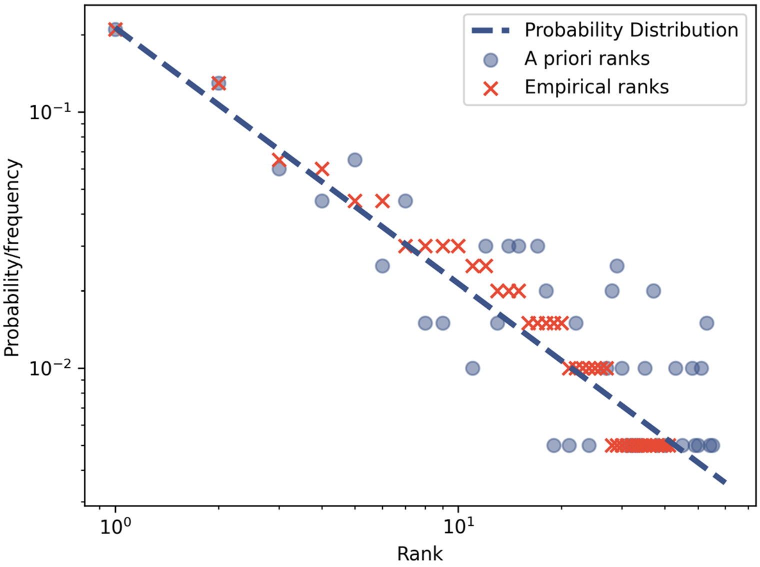

Figure 2. Difference between distributions with probability and empirical ranks. Data was generated from an underlying power law probability distribution with exponent $\lambda = 1$, number of possible events $W = 60$ and $N = 200$ samples. The dotted blue line shows the probability distribution. The blue circles show the sampled event frequencies with a priori known probability ranks. The red crosses show the empirical rank-frequency distribution from the same data. There is a significant difference between the two distributions. The current estimators are designed to fit data with a priori known ranks, not empirical ranks.

Recently Clauset et al.'s estimator has been shown, empirically, to be biased for some rank-frequency distributions[2,9,10]. In particular, Clauset et al.'s method over-estimates exponents with rank-frequency data generated from known power law probability distributions with exponents below about 1.5[10] (Fig. 1). The problem is related to low sampling in the tail[9,10], so that the observed empirical ranks tend to "bunch up" above the line of the true probability distribution before decaying sharply at the end of the observed tail (Fig. 2). To our knowledge this bias has not been adequately explained or solved.

- In 2014 Piantadosi et al.[2] explained the problem and suggested splitting a corpora and calculating ranks of words from one part of the split and frequencies from the other, breaking the correlation of errors. However the method does not take into account uncorrelated errors in the ranks. In particular, the empirical ranks of events in the tail will almost certainly be lower than the actual ranks in the probability distribution as many events in the tail will not be observed at all.
- Hanel et al.[10] identified the problem and suggested using a finite set of events instead of Clauset et al.'s unbounded event set[8]. This gives more accurate results in the limited case that the number of possible events, W, is finite and known[10]. Often W is not known and the choice of W can substantially change the results. With Zipf's law in language, W represents the writer's vocabulary and is usually modelled as unbounded[2,8,12]. This seems appropriate given that Heaps' Law suggests that the number of unique words in a document continues to rise indefinitely as the document length increases[14].
- In 2019 Corral et al.[9] examined the problem and explored a technique of transforming the data to a distribution of frequencies representation, $f(n)$, which is also a power law type distribution that they call the Zipf's law for sizes[9]. This distribution has an a priori known independent variable of frequency sizes, so the bias does not apply to this representation. However there is still difficulty in estimating the rank-frequency exponent, as a power law in the rank-frequency distribution, $n(r_e)$, will only approximately map to a power law in the distribution of frequencies, $f(n)$, for real-world sample sizes[9].

Overall these ad-hoc methods can remove the bias to some extent but not completely. The methods also introduce a host of somewhat arbitrary choices for the researcher to resolve.

We derive a new maximum likelihood estimator that does not make the false assumption that the empirical ranks, r_e, are equivalent to the probability ranks, r_p. The new estimator considers all the possible ways that the events could be ranked in the underlying probability distribution to generate the observed empirical data. Unfortunately this new likelihood function is computationally intractable for all but the smallest data sets. In order to estimate parameters for larger data sets, we turn to approximate Bayesian computation (ABC), a method that is designed for situations where likelihood functions cannot be computed[15]. We show that this method has much lower bias than Clauset et al.'s estimator for rank-frequency data generated from simple power laws. We further explore two different implementations of ABC and find that they give different results when applied to word distributions in books because ABC and Clauset et al.'s method both assume an underlying power law probability model, while natural language arises from a more complex model. We suggest that this false assumption means that maximum likelihood estimation with simple models will always have some arbitrary bias when studying rank-frequency data in natural language, including ABC and Clauset et al.'s method.

Model

Likelihood function: general case with no a priori ordering.

A vector of data, $\boldsymbol{d} = [d_1, d_2, \dots d_N]$, represents N observations of a random variable X. Each of these observations are one of a discrete set of W events, with no a priori ordinality. An example is words in a book.

We can transform the vector $\boldsymbol{d}$ to counts of each event, ordered from most to least frequent, $\boldsymbol{n} = [n(x_{(1)}), n(x_{(2)}), \dots, n(x_{(W)})]$. $\boldsymbol{n}(x_{(r_e)})$ represents the count of the r_eth most common event, where r_e is the event's ranking in the empirical frequency distribution. For ease of notation we will refer to $\boldsymbol{n}(x_{(r_e)})$ as $\boldsymbol{n}(r_e)$.

We assume a simple model where each of these events has some unknown fixed probability of being observed, $p(x_{r_p}) = Pr(X = x_{r_p})$, where r_p is the event's rank in the underlying probability distribution.

The key insight is that given an event's empirical rank, we do not know that event's rank in the underlying probability distribution. We can describe the mapping of events from the data generating probability ranking to the empirical ranking with a vector $\boldsymbol{s}$, so that $\boldsymbol{s}(r_p) = r_e$. For example $\boldsymbol{s} = [2, 1, 3]$ would mean that the second most probable event was observed empirically the most number of times, the most probable event was seen the second most number of times, and the third most likely seen third most. For any valid mapping, $\boldsymbol{s}$ must be a permutation of the integers from 1 to W. Figure 3 shows an example mapping.

We assume that the probability distribution is parameterised by $\boldsymbol{\theta}$. Considering Bayes' rule

$$p(\boldsymbol{\theta}|\boldsymbol{n}) = \frac{p(\boldsymbol{n}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{n})} \tag{3}$$

The likelihood can be written as (ignoring constants of proportionality)

$$p(\boldsymbol{n}|\boldsymbol{\theta}) = \prod_{r_e=1}^{W} p(x_{(r_e)})^{\boldsymbol{n}(r_e)} \tag{4}$$

This likelihood equation is in terms of the events' empirical rank, r_e, whereas the underlying probability model is in terms of probability rank, r_p. To convert the likelihood to be in terms of r_p we condition on the mapping vector, $\boldsymbol{s}$.

$$p(\boldsymbol{n}|\boldsymbol{\theta}, \boldsymbol{s}) = \prod_{r_p=1}^{W} p(x_{r_p})^{\boldsymbol{n}(\boldsymbol{s}(r_p))} \tag{5}$$

Using the law of total probability we sum over all possible mappings of probability rankings onto empirical rankings. $S(W)$ is the set of all possible permutations of the numbers 1 to W, known as the symmetric group.

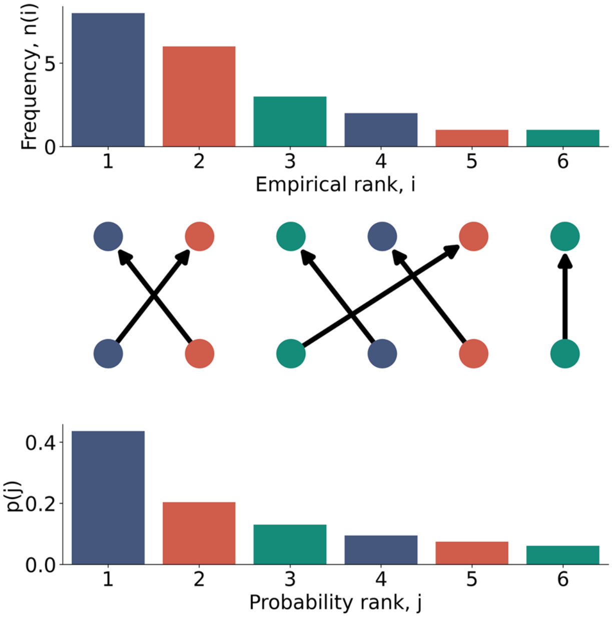

Figure 3. An example mapping from probability to empirical ranks. The observed data $\boldsymbol{n} = [8, 6, 3, 2, 1, 1]$ can arise from any valid permutation of events from the probability distribution. Here the permutation is $\boldsymbol{s} = [2, 1, 5, 3, 4, 6]$. The 1st most likely event is observed the second most times ($\boldsymbol{s}[1] = 2$), etc. The likelihood of the data given this permutation is $p(\boldsymbol{n}|\boldsymbol{s}, \boldsymbol{\theta}) = p_1^6 p_2^8 p_3^1 p_4^3 p_5^2 p_6^1$.

$$p(\boldsymbol{n}|\boldsymbol{\theta}) = \sum_{\boldsymbol{s} \in S(W)} \prod_{r_p=1}^{W} p(x_{r_p})^{\boldsymbol{n}(\boldsymbol{s}(r_p))} \tag{6}$$

Equation (6) is the likelihood for any data that represents observations of discrete events, where the events have no a priori ordering in relation to the underlying model. The equation generalises to $W \to \infty$, suitable to describe models with unbounded event sets, as is the case in many Zipf type models.

Likelihood function: power laws with no a priori ordering.

A common model applied to rank-frequency distributions is the power law, used by Zipf in his study of words[1]. A power law probability distribution is of the form

$$p(x_{r_p}) = \frac{r_p^{-\lambda}}{Z_\lambda} \tag{7}$$

where λ is the power law exponent, Z_λ is a normalising factor. We use the simplest form of Zipf's law for ease of analysis. The method described here can be used with other models such as the Zipf–Mandelbrot law[16]. The normalising factor is:

$$Z_\lambda = \sum_{r_p=1}^{W} r_p^{-\lambda} \tag{8}$$

W is the number of possible events. In the limit $W \to \infty$, Z_λ becomes the Riemann zeta function, $\zeta(\lambda)$[8]. Considering Eq. (6), the likelihood can be written as

$$\mathcal{L}(\lambda|\boldsymbol{n}) = \sum_{\boldsymbol{s} \in S(W)} \prod_{r_p}^{W} \left(\frac{r_p^{-\lambda}}{Z_\lambda} \right)^{\boldsymbol{n}(\boldsymbol{s}(r_p))} \tag{9}$$

And the differential of the likelihood with respect to λ is

$$\frac{\partial}{\partial \lambda} \mathcal{L}(\lambda|\boldsymbol{n}) = \sum_{\boldsymbol{s} \in S(W)} \left(\left(\frac{N Z'_\lambda}{Z_\lambda} + \sum_{r_p}^{W} \boldsymbol{n}(\boldsymbol{s}(r_p)) ln(r_p) \right) \times \prod_{r_p}^{W} \left(\frac{r_p^{-\lambda}}{Z_\lambda} \right)^{\boldsymbol{n}(\boldsymbol{s}(r_p))} \right) \tag{10}$$

Z'_λ is the differential of the normalising factor with respect to λ. To find the maximum likelihood estimator, we can use numerical methods to either (a) maximise Eq. (9) or (b) find the root of Eq. (10) (Fig. 4).

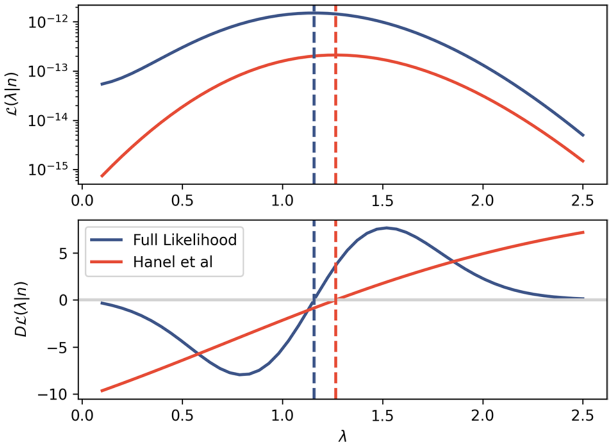

Figure 4. Likelihood functions of the full likelihood (blue) and only the leading term (red). Both likelihoods are calculated for the data $\boldsymbol{n} = [10, 3, 3, 2, 1, 1]$. The leading term of the full likelihood is equivalent to the likelihood function as defined by Hanel et al.[10], which is adapted for finite event sets from Clauset et al.'s estimator[8]. The top figure shows the full likelihood compared to Hanel et al.'s likelihood, with the maximum likelihood estimators shown as dashed lines. The bottom figure shows the differential of the likelihood functions. The form of the differential of the full likelihood is markedly different to only the first term. There is a substantial difference in the maximum likelihood estimator, with the Hanel et al. estimator giving $\hat{\hat{\lambda}} = 1.27$ and the full estimator giving $\hat{\hat{\lambda}} = 1.16$.

The prevailing estimators from the literature (often implicitly) assume that the empirical ranks match the probability ranks[2,8,12], so that they only consider the leading term in the main sum in both Eqs. (9) and (10) (associated with the identity permutation $\boldsymbol{s}_I = [1, 2, \ldots, W]$). This is the source of the bias in the existing estimators.

The number of terms in the likelihood function (Eq. 6) scales as $O(W!)$, so that naive computation of the likelihood is impractical even at $W \approx 10$. The computation can be shown to be equivalent to the computation of the permanent of a matrix with entries $a_{ij} = p(x_j)^{\boldsymbol{n}(i)}$. The best known algorithm for exactly computing the permanent of a matrix is Ryser's algorithm[17,18] with complexity $O(W2^W)$. This is computationally intractable for real world data sets such as text corpora with vocabularies of $W > 1000$. A more in-depth discussion on the computational complexity can be found in the Supplementary Information.

Approximate Bayesian computation. Approximate Bayesian computation is a technique for approximating posterior distributions without calculating a likelihood function[19–21]. Instead, we assume a model, $\mathcal{M}$, simulate data, $\boldsymbol{n}_i$, from possible parameters, λ_i, and observe how close that simulated data is to the empirical data using a distance measure $\rho(\boldsymbol{n}_i, \boldsymbol{n}_{obs})$[19,21]. The ABC rejection algorithm is based upon the principle that we can approximate the actual posterior by estimating the probability of λ given that the data is within some small tolerance, ϵ, of the observed empirical data[19,22]. This assumes that the model, $\mathcal{M}$, is a good representation of the actual data generating process.

$$p(\lambda|\boldsymbol{n} = \boldsymbol{n}_{obs}, \mathcal{M}) \approx p(\lambda|\rho(\boldsymbol{n}, \boldsymbol{n}_{obs}) < \epsilon, \mathcal{M}) \tag{11}$$

$$p(\lambda|\rho(\boldsymbol{n}, \boldsymbol{n}_{obs}) < \epsilon, \mathcal{M}) = \frac{p(\rho(\boldsymbol{n}, \boldsymbol{n}_{obs}) < \epsilon|\lambda, \mathcal{M})p(\lambda|\mathcal{M})}{p(\rho(\boldsymbol{n}, \boldsymbol{n}_{obs}) < \epsilon|\mathcal{M})} \tag{12}$$

The ABC rejection algorithm begins by sampling parameter values from the prior. For each of these parameter values, data is then generated from the model and tested on the condition $\rho(\boldsymbol{n}_i, \boldsymbol{n}_{obs}) < \epsilon$[19]. With enough samples, the density of successful parameters will approximate the right hand side of Eq. (12), and an approximation for the posterior distribution[19]. If we use a uniform prior then this will be a proportional estimate to the likelihood.

An ideal distance measure, $\rho(\boldsymbol{n}_i, \boldsymbol{n}_{obs})$, would involve comparing Bayesian sufficient summary statistics from the data[21]. Usually in practice Bayesian sufficiency cannot be achieved[19,21], and some information will be lost so that the approximation of the posterior includes some error[19]. A common technique is to summarise the data sets with summary statistics, $\boldsymbol{S}(\boldsymbol{n})$, and define the distance as the difference between those, $\rho(\boldsymbol{n}_i, \boldsymbol{n}_{obs}) = \boldsymbol{S}(\boldsymbol{n}_i) - \boldsymbol{S}(\boldsymbol{n}_{obs})$[15,19,21]. Recently the Wasserstein distance, a metric between distributions, has been shown to work well as a distance measure[23]. This is a principled approach that avoids the difficult selection of summary statistics[23], and this is the measure that we use here.

The ABC rejection algorithm requires a small tolerance in order to find a good estimate for the posterior[22]. This in turn requires a high density of samples in order to have enough successful parameters to build the posterior approximation. To sample at a high density across a reasonable parameter space with a uniform prior would be prohibitively computationally expensive. Instead, we use population Monte Carlo to sample from a proposal distribution that focuses on areas of high posterior probability while avoiding areas of negligible probability[24]. At each time step, the results are weighted using principles from importance sampling to account for the fact that

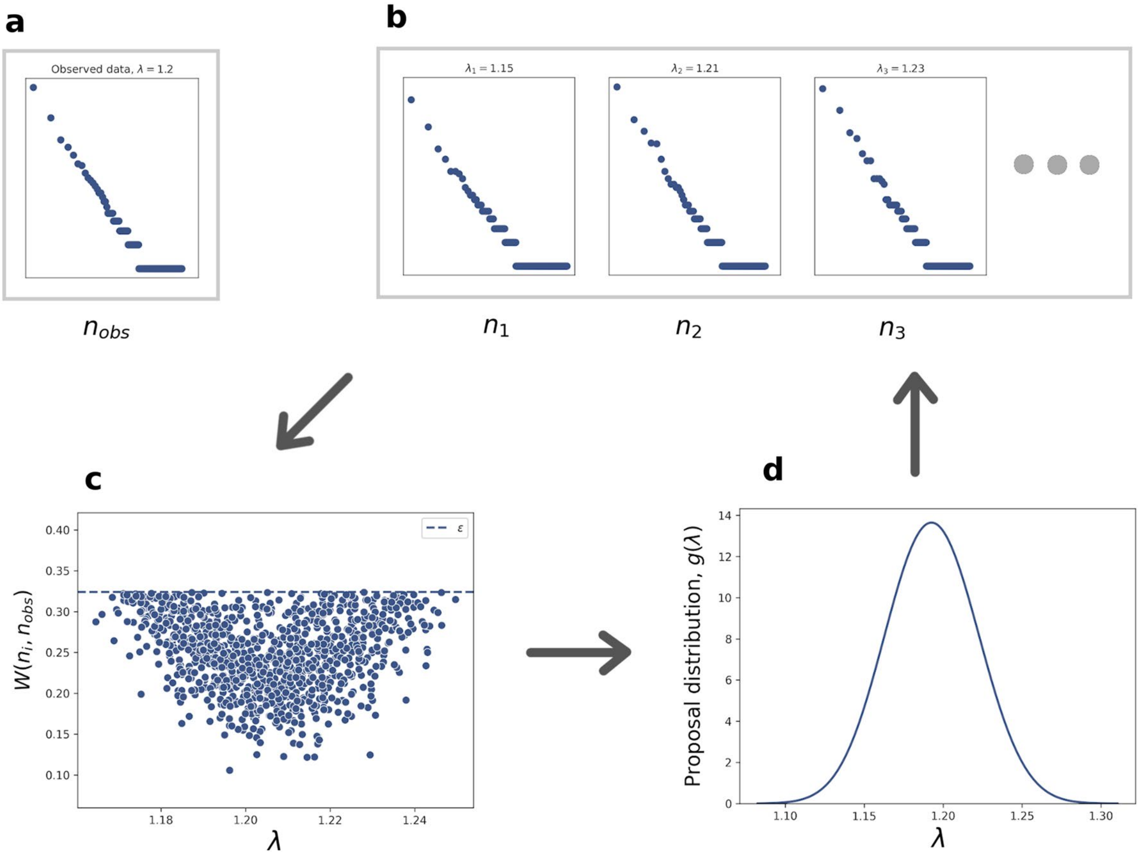

Figure 5. Approximate Bayesian computation with population Monte Carlo (ABC-PMC). (**a**) Given the observed data. (**b**) Particles are generated from a proposal distribution and data is simulated for each particle. For each particle, the Wasserstein distance is measured between the simulated data and the observed data. (**c**) This is repeated until *nParticles* samples are generated with Wasserstein distance within a tolerance ϵ. (**d**) A new proposal distribution is generated by a weighted kernel density estimate on the accepted particles, with a weighting based on importance sampling principles. A new tolerance is set based upon a proportion of *survivalFraction* particles with the smallest distances found in this time step. This is repeated for a given number of generations. The final successful particles are used to generate an approximation of the posterior distribution using a weighted kernel density estimate. Figure adapted in part from[19] and[21].

we are sampling from the proposal distribution instead of the prior[24]. This algorithm, adapted from[25], is shown in Algorithm 1 and Fig. 5 (the 2 parameter algorithm is equivalent, with the variance replaced by a covariance matrix). The parameters in the algorithm were set following trial and error to balance computation time and accuracy.

We also investigated an alternative approach to approximate Bayesian computation known as ABC regression. Instead of the Wasserstein distance, we used the mean of the log transformed event counts as a summary statistic with this method. Full details are in the Supplementary Information.

Algorithm 1: Approximate Bayesian Computation Population Monte Carlo Zipf's Law

```
Input: The observed data $\boldsymbol{n} = [n_1, n_2, \ldots, n_W]$,
       $\theta_{min} \leftarrow 1.001, \theta_{max} \leftarrow 3, survivalFraction \leftarrow 0.4, nParticles \leftarrow 256, nGenerations \leftarrow 10$
Output: Maximum likelihood estimator $\hat{\theta}$
$priorDist \leftarrow uniformDist(\theta_{min}, \theta_{max})$
$nData \leftarrow sum(\boldsymbol{n})$
$tolerance \leftarrow \infty$
$proposalDist \leftarrow priorDist$
for $g \leftarrow 1$ to $nGenerations$ do
    $\theta s \leftarrow array()$
    $ds \leftarrow array()$
    $weights \leftarrow array()$
    for $i \leftarrow 1$ to $nParticles$ do
        $hit \leftarrow FALSE$
        while $!hit$ do
            $\theta \leftarrow proposalDist.sample()$
            if $\theta_{min} \leq \theta \leq \theta_{max}$ then
                $z \leftarrow generateData(\theta, nData)$
                $d \leftarrow wassersteinDistance(n, z)$
                if $d \leq tolerance$ then
                    $\theta s[i] \leftarrow \theta$
                    $ds[i] \leftarrow d$
                    $weights[i] \leftarrow priorDist.evaluate(\theta)/proposalDist.evaluate(\theta)$
                    $hit \leftarrow TRUE$
    $tolerance \leftarrow getTolerance(ds, survivalFraction)$
    $var \leftarrow weightedVariance(\theta s, weights)$
    $proposalDist \leftarrow KDE(\theta s, weights, bandwidth = sqrt(2 \times var))$
$posterior \leftarrow KDE(\theta s, weights, bandwidth = sqrt(var))$
$\hat{\theta} \leftarrow max(posterior)$
return $\hat{\theta}$
```

ABC results

Approximate Bayesian computation with Zipf distributions.

Rank-frequency data was generated (N =10,000) from an unbounded power law with exponents ranging from 1 to 2. For each generated data set, the exponent was estimated using a) Clauset et al.'s estimator and b) ABC-PMC with the Wasserstein distance. This was repeated 100 times to find the mean bias and variance. The ABC method has much lower bias and similar variance to Clauset et al.'s method, (Fig. 6).

We also investigated how the bias changes with varying sample size. Rank-frequency data was generated with $\lambda = 1.1$ and varying sample size up to N =1,000,000. Clauset et al.'s estimator shows positive bias at all values of N, although it decreases with large N. ABC shows much lower bias for all values of N. The variance of ABC is higher for $N \lessapprox 1000$. Overall the variance is still very low, and is insignificant compared to the positive bias showed by Clauset et al.'s estimator (Fig. 7).

In addition to the results shown here, we explored a variation of the algorithm using ABC rejection with the mean of the logged event counts as a summary statistic. This method has similarly low bias and variance as the results shown here. See the Supplementary Information for full details.

Approximate Bayesian computation with Zipf–Mandelbrot model.

The Zipf–Mandelbrot law is a modification of Zipf's law derived by Mandelbrot that accounts for a departure from a strict power law in the head of the rank-frequency distribution[16].

$$p(r_p) \propto (r_p + q)^{-\lambda} \quad q \in [0, 1, 2 \ldots] \tag{13}$$

We tested the ABC PMC algorithm with this 2 parameter model. The algorithm is of the same form as Algorithm 1, with the variance replaced with a covariance matrix. The algorithm is demonstrated with one generated data set with q =4, λ=1.2 and N =100,000. ABC PMC performs well, with close estimates to the true parameters (see Fig. 8). The approximated likelihood function gives negligible probability for q =0, suggesting that the algorithm can discriminate between data generated from Zipf's law and the Zipf–Mandelbrot law.

Analysis of books.

Both Clauset et al.'s method and the approximate Bayesian computation method described here assume a Zipfian data generating model. We have demonstrated that ABC-PMC with the Wasserstein distance works well for data generated from a known power law, with much lower bias than Clasuet

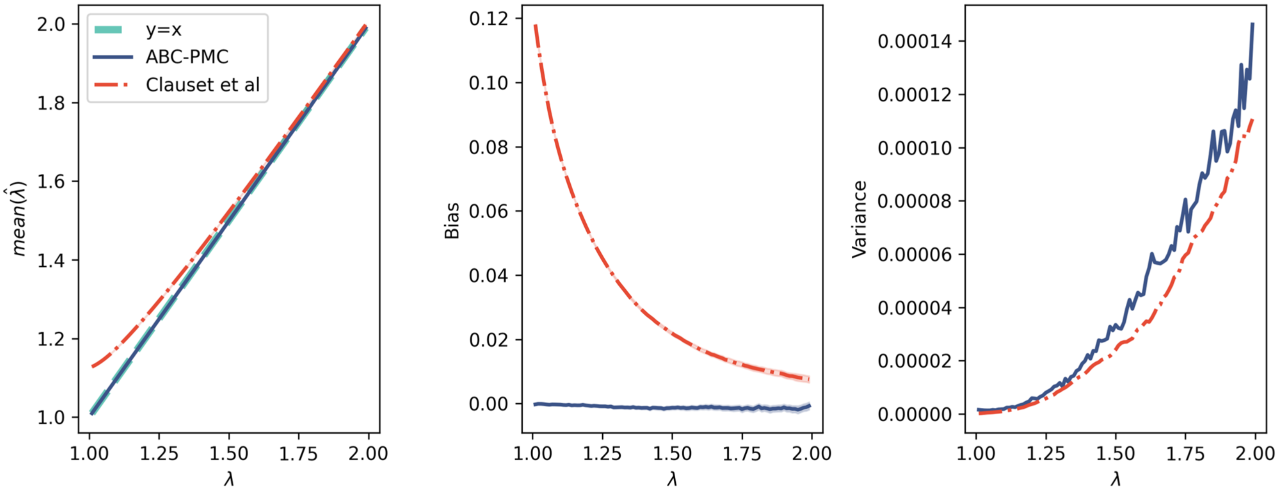

Figure 6. Bias in ABC (solid blue) vs Clauset et al.'s estimator (dashed red) for unbounded power laws. For each of 100 values of λ between 1.01 and 2, rank-frequency data (N =10,000) was generated by sampling an unbounded power law. This was run 100 times. The left figure shows the known λ and the mean estimated λ. The centre figure shows the mean bias, with a 68% confidence interval shaded. The right figure shows the variance of the estimators. The ABC estimator has much lower bias and similar variance to Clauset et al.'s estimator.

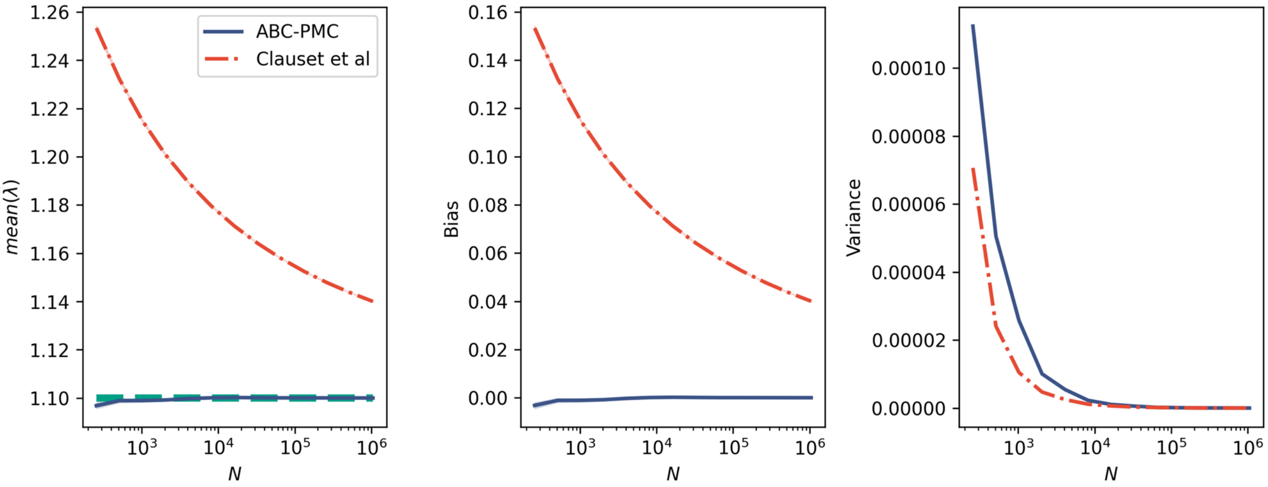

Figure 7. Bias in ABC (solid blue) vs Clauset et al.'s estimator (dashed red) for unbounded power laws. Rank-frequency data was generated for $\lambda = 1.1$ with varying sizes, N. This was run 100 times. The left figure shows the known λ against the mean estimated λ. The centre figure shows the mean bias, with a 68% confidence interval shaded. The right figure shows the variance of the estimators. The bias is much lower with ABC. The ABC estimator has higher variance than Clauset et al. at low N, although the variance is still very low.

et al.'s method. In the Supplementary Information, we also describe an ABC regression method using the mean log of the word counts that has similar low bias when applied to data from a power law distribution.

It is reasonable to suggest that natural language is a more complex process than drawing words from a power law probability distribution. Indeed, deep learning language models like GPT-3 use billions of parameters[26]. As such, models that assume Zipfian data generating models are not necessarily suitable for analysing language. To demonstrate the problem, we analysed books using (a) Clauset et al.'s method, (b) ABC-PMC with the Wasserstein distance (c) ABC regression with the mean of the log transformed word counts as a summary statistic (Table 1). All of the books were downloaded from Project Gutenberg[27]. Each text sample was first "cleaned" by removing all punctuation, replacing numbers with a # symbol, and converting all text to lowercase. The word frequencies were then counted.

The two forms of ABC give different results, which bracket the results of the Clauset et al. estimator. This does not imply that the Clauset et al. is the best approximator as we show above that it is biased upwards. What these present results indicate is that there is no correct "ground truth" because the assumed underlying models are wrong.

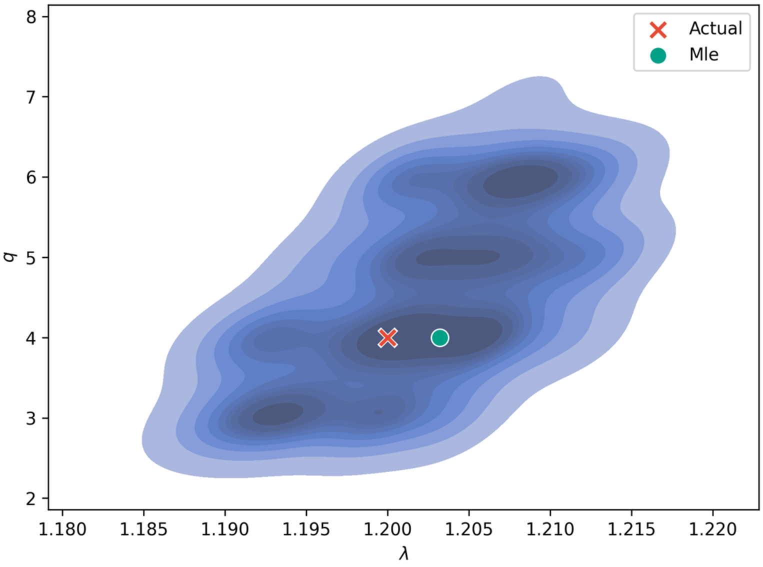

Figure 8. Results of ABC-PMC for the Zipf–Mandelbrot law with data generated with known exponent $\lambda = 1.2$ and $q = 4$ (red cross) with N =100,000 words. The likelihood function (darker blue regions have higher likelihood) was approximated using a kernel density estimate. The mode of the KDE gives the maximum likelihood estimate (green circle). The estimator correctly identifies q and is close to the correct exponent λ.

Book	Clauset et al.	ABC PMC with Wasserstein	ABC regression with mean log
Moby dick	1.19	1.25	1.16
A tale of two cities	1.21	1.27	1.17
Alice in Wonderland	1.22	1.25	1.18
Chronicles of London	1.19	1.20	1.15
Ulysses	1.18	1.22	1.14

Table 1. Comparision of estimators of Zipf's law in books

Discussion

We have demonstrated that the prevailing Zipf's law maximum likelihood estimators for rank-frequency data are biased due to an inappropriate likelihood function. This bias is particularly strong in the range of natural language, with exponents close to 1. The correct likelihood function is intractable. We have presented one approach to overcoming this bias using a likelihood-free method of approximate Bayesian computation. The ABC method is shown to work well with data generated from actual power law distributions, with lower bias than Clauset et al.'s estimator.

ABC works well in an idealised situation where the true model is known. However when applied to analysing books, the two ABC approaches that we explored give very different estimates for the Zipf exponents. The Zipfian approaches we investigate all assume a simple bag of words probability model, whereas our results on books indicate that natural language generation is a more complex process–otherwise the two ABC methods would converge. The ABC algorithms are searching a parameter space for the closest model based on the distance measure. This works well when the parameter space includes the true data generating process. But with natural language the assumed simple Zipf model is wrong so there is no "correct" location in the parameter space (or the "correct" location is outside the parameter space). Different distance measures will prejudice different aspects of the observed data and so arrive at different estimates. This bias is arbitrary in nature and there seems to be no reasonable way to decide which distance measure is "correct". The error lies in the assumption of an incorrect data generating model. This problem applies to ABC and Clauset et al.'s estimator, and seems to be inherent in applying maximum likelihood estimation using simple models to describe power laws in natural language.

Zipf's law for word types is an empirical relationship between frequencies of words and ranks in that frequency distribution. The difficulty arises when a probabilistic model is used to describe the mechanism that is generating this relationship, when the actual mechanism is more complex. The main aim of this publication is to clearly show that Clauset et al.'s estimator is strongly biased for rank-frequency data. The correct likelihood function provides an unbiased framework that works well when the underlying data generating process is known. This does not appear to be the case for natural language. Graphical methods may therefore be more suitable to study Zipf's law when investigating the empirical relationship between ranks and frequencies (Eq. 1) and not the probability distribution (Eq. 2). All Zipf estimators have some bias and the best choice will depend on the specific application.

The scripts and data used here are available at the repository https://github.com/chasmani/PUBLIC_bias_in_zipfs_law_estimators. That repository includes the approximate Bayesian computation algorithm as well as implementations of other estimators from the literature.

Received: 12 August 2020; Accepted: 12 January 2021
Published online: 27 August 2021

References

1. Zipf, G. K. *Human Behavior and the Principle of Least Effort.* (Addison-wesley press, 1949).
2. Piantadosi, S. T. & Piantadosi, S. T. Zipf's word frequency law in natural language: A critical review and future directions. *Psychon. Bull. Rev.* **21**, 1112–1130. https://doi.org/10.3758/s13423-014-0585-6 (2014).
3. Ferrer, R. & Cancho, R. The variation of Zipf's law in human language. *Eur. Phys. J. B* **44**, 249–257. https://doi.org/10.1140/epjb/e2005-00121-8 (2005).
4. Moreno-Sánchez, I., Font-Clos, F. & Corral, Á. Large-scale analysis of Zipf's law in english texts. *PLoS ONE* **11**, e0147073. https://doi.org/10.1371/journal.pone.0147073 (2016).
5. Montemurro, M. A. & Zanette, D. H. New perspectives on zipf's law in linguistics: From single texts to large corpora. *Glottometrics* **4**, 87–99 (2002).
6. Shannon, C. E. Prediction and entropy of printed english. *Bell Syst. Tech. J.* **30**, 50–64 (1951).
7. Newman, M. E. Power laws, pareto distributions and zipf's law. *Contemp. Phys.* **46**, 323–351 (2005).
8. Clauset, A., Shalizi, C. R. & Newman, M. E. Power-law distributions in empirical data. *SIAM Rev.* **51**, 661–703 (2009).
9. Corral, A., Serra, I. & Ferrer-i Cancho, R. The distinct flavors of zipf's law in the rank-size and in the size-distribution representations, and its maximum-likelihood fitting. arXiv preprint arXiv:1908.01398 (2019).
10. Hanel, R., Corominas-Murtra, B., Liu, B. & Thurner, S. Fitting power-laws in empirical data with estimators that work for all exponents. *PLoS ONE* **12**, e0170920. https://doi.org/10.1371/journal.pone.0170920 (2017).
11. Goldstein, M. L., Morris, S. A. & Yen, G. G. Problems with fitting to the Power-law distribution. *Eur. Phys. J. B*https://doi.org/10.1140/epjb/e2004-00316-5 *(2004).*
12. Bauke, H. Parameter estimation for power-law distributions by maximum likelihood methods. *Eur. Phys. J. B* **58**, 167–173. https://doi.org/10.1140/epjb/e2007-00219-y (2007).
13. Seal, H. The maximum likelihood fitting of the discrete pareto law. *J. Inst. Actuar.* **1886–1994**(78), 115–121 (1952).
14. Heaps, H. S. *Information Retrieval, Computational and Theoretical Aspects* (Academic Press, 1978).
15. Beaumont, M. A. Approximate Bayesian computation in evolution and ecology. *Annu. Rev. Ecol. Evol. Syst.* **41**, 379–406. https://doi.org/10.1146/annurev-ecolsys-102209-144621 (2010).
16. Mandelbrot, B. An informational theory of the statistical structure of language. *Commun. Theory* **84**, 486–502 (1953).
17. Ryser, H. J. *Combinatorial Mathematics*, vol. 14 (American Mathematical Soc., 1963).
18. Glynn, D. G. The permanent of a square matrix. *Eur. J. Comb.* **31**, 1887–1891. https://doi.org/10.1016/j.ejc.2010.01.010 (2010).
19. Sunnåker, M. *et al.* Approximate Bayesian computation. *PLoS Comput. Biol.* **9**, e1002803. https://doi.org/10.1371/journal.pcbi.1002803 (2013).
20. Beaumont, M. A., Zhang, W. & Balding, D. J. Approximate Bayesian computation in population genetics. *Genetics* **162**, 2025–2035 (2002).
21. Csilléry, K., Blum, M. G. B., Gaggiotti, O. E. & François, O. Approximate Bayesian computation (ABC) in practice. *Trends Ecol. Evol.* **25**, 410–418. https://doi.org/10.1016/j.tree.2010.04.001 (2010).
22. Sisson, S. A., Fan, Y. & Tanaka, M. M. Sequential Monte Carlo without likelihoods. *Proc. Natl. Acad. Sci. U.S.A.* **104**, 1760–1765. https://doi.org/10.1073/pnas.0607208104 (2007).
23. Bernton, E., Jacob, P. E., Gerber, M. & Robert, C. P. Approximate bayesian computation with the wasserstein distance. arXiv preprint arXiv:1905.03747 (2019).
24. Cappé, O., Guillin, A., Marin, J. M. & Robert, C. P. Population Monte Carlo. *J. Comput. Graph. Stat.* **13**, 907–929. https://doi.org/10.1198/106186004X12803 (2004).
25. Beaumont, M. A., Cornuet, J.-M., Marin, J.-M. & Robert, C. P. Adaptive approximate Bayesian computation. *Biometrika* **96**, 983–990. https://doi.org/10.1093/biomet/asp052 (2009).
26. Brown, T. B. *et al.* Language models are few-shot learners. arXiv preprint arXiv:2005.14165 (2020).
27. Project Gutenberg (2020). [Online; accessed 16. Jul. 2020].

Acknowledgements

The study was funded by the EPSRC grant for the Mathematics for Real-World Systems CDT at Warwick (Grant No. EP/L015374/1). T.T.H. was supported on this work by the Royal Society Wolfson Research Merit Award (WM160074) and a Fellowship from the Alan Turing Institute.

Author contributions

C.P. conceived of the presented idea and carried out the analyses. T.T.H. supervised C.P. and offered guidance, suggestions and support throughout. All authors reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-021-96214-w.

Correspondence and requests for materials should be addressed to C.P.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.