Journal of Statistical Software

August 2020, Volume 94, Issue 11. doi: 10.18637/jss.v094.i11

deconvolveR: A G-Modeling Program for Deconvolution and Empirical Bayes Estimation

Balasubramanian Narasimhan
Stanford University

Bradley Efron
Stanford University

Abstract

Empirical Bayes inference assumes an unknown prior density $g(\theta)$ has yielded (unobservables) $\Theta_1, \Theta_2, \ldots, \Theta_N$, and each Θ_i produces an independent observation X_i from $p_i(X_i|\Theta_i)$. The marginal density $f_i(X_i)$ is a convolution of the prior g and p_i. The Bayes deconvolution problem is one of recovering g from the data. Although estimation of g – so called g-modeling – is difficult, the results are more encouraging if the prior g is restricted to lie within a parametric family of distributions. We present a deconvolution approach where g is restricted to be in a parametric exponential family, along with an R package **deconvolveR** designed for the purpose.

Keywords: Bayes deconvolution, g-modeling, empirical Bayes, missing species, R package **deconvolveR**.

1. Introduction

Modern scientific technology excels at the production of large data sets composed of a great many small estimation problems. A microarray experiment, for example, might produce N one-dimensional Normal theory estimates X_i,

$$X_i \sim \mathcal{N}(\Theta_i, 1), \qquad i = 1, 2, \ldots, N, \tag{1}$$

with the estimation of the Θ_i's being the goal. This was the case for the prostate cancer study pictured in Figure 2.1 of Efron (2010), where there were $N = 6,033$ genes, with X_i measuring a standardized difference between patients and controls for the ith gene.

A Bayesian analysis begins with a prior density $g(\theta)$ for the Θ_i. Inference is based on the posterior density of Θ_i given $X_i = x$,

$$g(\theta \mid x) = g(\theta)p(x \mid \theta)/f(x); \tag{2}$$

here $p(x \mid \theta)$ is the density of X given $\Theta = \theta$, and $f(x)$ is the marginal density of X,

$$f(x) = \int_{-\infty}^{\infty} g(\theta) p(x \mid \theta)\ d\theta. \tag{3}$$

In the setting of Equation 1 $p(x \mid \theta)$ is $\varphi(x - \theta)$, with φ the standard Normal density $\exp(-x^2/2)/\sqrt{2\pi}$.

What if we don't know the prior density $g(\theta)$? Empirical Bayes methods attempt to estimate $g(\theta)$ from the observed sample $\boldsymbol{X} = (X_1, X_2, \ldots, X_N)$. An estimate $\hat{g}(\cdot)$ then produces posterior approximations $\hat{g}(\theta \mid x)$ from Equation 2. Both $\hat{g}(\theta)$ and $\hat{g}(\theta \mid x)$ can be of interest in applied problems.

In the Normal model above, $f(x)$ is the *convolution* of $g(\theta)$ with a standard $\mathcal{N}(0,1)$ density. The empirical Bayes task is one of *deconvolution*: using the observed sample $\boldsymbol{X}$ from $f(x)$ to estimate $g(\theta)$. This can be a formidable job. Convergence rates of $\hat{g}$ to g are notoriously slow in the general framework where $g(\theta)$ can be anything at all (Carroll and Hall 1988). Efron (2016) showed that parametric models, where $g(\theta)$ is assumed to lie in a known exponential family, allow reasonably efficient and practical estimation algorithms. This is the "g-modeling" referred to in our title.

Empirical Bayes deconvolution and estimation does not require the Normal model. We might, for example, have

$$X_i \sim \text{Poisson}(\Theta_i), \tag{4}$$

or X_i given Θ_i Binomial, etc., the only requirement being a known specification of the distribution $p(x \mid \theta)$ for X_i given Θ_i. The "Bayes deconvolution problem" is a general name for estimating $g(\theta)$ in Equation 3 given a random sample from $f(x)$.

Empirical Bayes applications have traditionally been dominated by f-modeling where the probability models for the marginal density $f(x)$, usually exponential families, are fit directly to the observed sample. Several packages for such estimation are available in R, particularly as part of the **Bioconductor** project (Gentleman *et al.* 2004). Package **siggenes** (Schwender 2020) implements the approach outlined in Efron (2010) for differential expression of genes; others, such as **baySeq** (Hardcastle 2019) and **edgeR** (Robinson, McCarthy, and Smyth 2010) use an empirical Bayes approach to estimate the parameter of a Negative Binomial prior or a dispersion parameter. Our package, **deconvolveR**, on the other hand, is devoted specifically to g-modeling.

Section 2 presents a brief review of g-modeling estimation theory, illustrated in Section 3 with a Poisson example relating to the "missing species problem", a classical empirical Bayes test case. The main content of this note appears in Section 4: a guide to a new R (R Core Team 2020) package **deconvolveR**, for the empirical Bayes estimation of $g(\theta)$ and $g(\theta \mid x)$. Package **deconvolveR** (Efron and Narasimhan 2020) is available from the Comprehensive R Archive Network (CRAN) at `https://CRAN.R-project.org/package=deconvolveR` and GitHub. Efron (2016) gives a full explanation of the theory and its implementation.

2. Empirical Bayes estimation theory

This section presents a condensed review of the empirical Bayes estimation theory in Efron (2014, 2016), emphasizing its application as carried out by the **deconvolveR** package of Section 4.

An unknown probability density $g(\theta)$ (possibly having discrete atoms) has yielded an unobservable random sample of independent realizations,

$$\Theta_i \overset{\text{ind}}{\sim} g(\theta) \qquad \text{for } i = 1, 2, \ldots, N. \tag{5}$$

Each Θ_i independently produces an observed value X_i according to a known family of probability densities $p(x \mid \theta)$,

$$X_i \overset{\text{ind}}{\sim} p(X_i \mid \Theta_i). \tag{6}$$

From the observer's point of view, the X_i are an independent and identically distributed (i.i.d.) sample from the marginal density $f(x)$,

$$f(x) = \int_{\mathcal{T}} p(x \mid \theta) g(\theta) \; d\theta, \tag{7}$$

$\mathcal{T}$ the sample space of the Θ_i. We wish to estimate $g(\theta)$ from the observed sample $\boldsymbol{X} = (X_1, X_2, \ldots, X_N)$.

The prior density $g(\theta)$ might in general be some mixture of discrete and continuous distributions. Here it will be convenient, both for explanation and computation, to assume Θ's sample space $\mathcal{T}$ to be finite and discrete,

$$\mathcal{T} = \left(\theta_{(1)}, \theta_{(2)}, \ldots, \theta_{(m)} \right). \tag{8}$$

A continuous formulation appears in Remark A1 of Efron (2016), but this is of no advantage in applications since Equation 8 can always be specified sufficiently finely to capture as much detail as is possible within the practical limitations of empirical Bayes inference. The support points $\theta_{(j)}$ in Equation 8 need not be equally spaced, and in fact are not in the Shakespeare example of Section 3. The only downside of unequal spacing is that some care is needed to give the correct visual impression when plotting $\hat{g}(\theta)$; see Figure 2 in Section 3. In particular, a "flat prior" won't have equal weights if the $\theta_{(j)}$'s are unequally spaced.

Choosing $\mathcal{T}$ can be a matter of some numerical experimentation, to see if a wider range or finer grid substantially changes the estimated prior $\hat{g}(\theta)$. For the Normal model (1), the range of $\mathcal{T}$ can be assumed smaller than that of the observed sample $X_1, X_2, \ldots, X_N$, though "not much smaller" is a good rule of thumb. There is no great penalty for increasing the number of support points m in Equation 8, but often no great gain either.

Similarly, $\mathcal{X}$, the sample space of the observations X_i, is assumed finite and discrete,

$$\mathcal{X} = \left(x_{(1)}, x_{(2)}, \ldots, x_{(n)} \right). \tag{9}$$

This is no restriction since $\mathcal{X}$ can be taken to be the entire order statistic of X_i values. (Or, for continuous situations like Equation 1, the X_i can be discretized by binning.)

In the discrete formulation of Equation 8, the prior $g(\theta)$ is represented by a vector $\boldsymbol{g} = (g_1, g_2, \ldots, g_m)^\top$. Likewise, the marginal $f(x)$ in Equation 7 has vector form $\boldsymbol{f} = (f_1, f_2, \ldots, f_n)^\top$. Both $\boldsymbol{g}$ and $\boldsymbol{f}$ have nonnegative components summing to 1. Letting

$$p_{kj} = \mathsf{P}\left\{ X = x_{(k)} \mid \Theta = \theta_{(j)} \right\}, \tag{10}$$

we define the $n \times m$ matrix $\boldsymbol{P} = (p_{kj})$. An advantage of the specifications in Equation 8 and Equation 9 is that the convolution-type relationship in Equation 3 between $g(\theta)$ and $f(x)$ reduces to matrix multiplication,

$$\boldsymbol{f} = \boldsymbol{P}\boldsymbol{g}. \tag{11}$$

The *count vector* $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)^\top$,

$$y_k = \#\{X_i = x_{(k)}\} \qquad \text{for } k = 1, 2, \ldots, n, \tag{12}$$

is a sufficient statistic for $\boldsymbol{g}$; it follows a multinomial distribution for n categories, N draws, probability vector $\boldsymbol{f}$,

$$\boldsymbol{y} \sim \text{mult}_n(N, \boldsymbol{f}). \tag{13}$$

G-modeling assumes that $g(\theta)$ is a member of a p-parameter exponential family on $\mathcal{T}$, expressed in the discrete formulation as

$$\boldsymbol{g}(\boldsymbol{\alpha}) = e^{\boldsymbol{Q}\boldsymbol{\alpha}}/c(\boldsymbol{\alpha}), \tag{14}$$

where $\boldsymbol{Q}$ is an $m \times p$ *structure matrix*, the default choice in **deconvolveR** being the natural spline basis $\text{ns}(\mathcal{T}, p)$; $\boldsymbol{\alpha}$ is the unknown p-dimensional natural parameter vector; $c(\boldsymbol{\alpha})$ is the divisor necessary to make $\boldsymbol{g}$ sum to 1. There is no deep mathematical reason for choosing splines, though their good behavior at the extremes of $\mathcal{T}$ helps reduce the volatility of $\hat{g}(\theta)$.

Coordinate-wise, Equation 14 says that

$$g_j(\boldsymbol{\alpha}) = e^{\boldsymbol{Q}_j^\top \boldsymbol{\alpha}}/c(\boldsymbol{\alpha}), \tag{15}$$

$\boldsymbol{Q}_j^\top$ the jth row of $\boldsymbol{Q}$, with

$$c(\boldsymbol{\alpha}) = \sum_{j=1}^{m} e^{\boldsymbol{Q}_j^\top \boldsymbol{\alpha}}. \tag{16}$$

The log likelihood function $l(\boldsymbol{\alpha})$ of $\boldsymbol{y}$ is

$$l(\boldsymbol{\alpha}) = \sum_{k=1}^{n} y_k \log f_k(\boldsymbol{\alpha}), \tag{17}$$

where $\boldsymbol{f}(\boldsymbol{\alpha}) = \boldsymbol{P}\boldsymbol{g}(\boldsymbol{\alpha})$. Define

$$w_{kj}(\boldsymbol{\alpha}) = g_j(\boldsymbol{\alpha}) \left\{ p_{kj}/f_k(\boldsymbol{\alpha}) - 1 \right\}, \tag{18}$$

and let $\boldsymbol{W}_k$ be the m-vector

$$\boldsymbol{W}_k(\boldsymbol{\alpha}) = \left[w_{k1}(\boldsymbol{\alpha}), w_{k2}(\boldsymbol{\alpha}), \ldots, w_{km}(\boldsymbol{\alpha}) \right]^\top, \tag{19}$$

for $k = 1, 2, \ldots, n$. (Note that $w_{kj}(\boldsymbol{\alpha})$ equals $g_{j/k}(\boldsymbol{\alpha}) - g_j(\boldsymbol{\alpha})$ where $g_{j/k}(\boldsymbol{\alpha})$ indicates the conditional probability of $\theta_{(j)}$ given $x_{(k)}$.)

The *score function* for $\boldsymbol{\alpha}$ then turns out to be

$$\begin{aligned} \dot{l}(\boldsymbol{\alpha}) &= \left(\frac{\partial l(\boldsymbol{\alpha})}{\partial \alpha_1}, \frac{\partial l(\boldsymbol{\alpha})}{\partial \alpha_2}, \ldots, \frac{\partial l(\boldsymbol{\alpha})}{\partial \alpha_p} \right)^\top \\ &= \boldsymbol{Q}^\top \boldsymbol{W}_+(\boldsymbol{\alpha}), \end{aligned} \tag{20}$$

where

$$\boldsymbol{W}_{+}(\boldsymbol{\alpha}) = \sum_{k=1}^{n} \boldsymbol{W}_k(\boldsymbol{\alpha}) y_k. \tag{21}$$

The maximum likelihood estimate (MLE) $\hat{\boldsymbol{\alpha}}$ is found by numerically maximizing $l(\boldsymbol{\alpha})$ or by solving $\dot{l}(\hat{\boldsymbol{\alpha}}) = 0$.

There is also a compact expression for the *Fisher information matrix* $\boldsymbol{\mathcal{I}}(\boldsymbol{\alpha}) = E_\alpha\{\dot{l}(\boldsymbol{\alpha})\dot{l}(\boldsymbol{\alpha})^\top\}$,

$$\boldsymbol{\mathcal{I}}(\boldsymbol{\alpha}) = N\boldsymbol{Q}^\top \left[\sum_{k=1}^{N} \boldsymbol{W}_k(\boldsymbol{\alpha}) f_k(\boldsymbol{\alpha}) \boldsymbol{W}_k(\boldsymbol{\alpha})^\top\right] \boldsymbol{Q}. \tag{22}$$

We could take $\boldsymbol{\mathcal{I}}(\hat{\boldsymbol{\alpha}})^{-1}$ as an estimate of covariance for $\hat{\boldsymbol{\alpha}}$. However a small amount of regularization greatly improves the stability of $\hat{\boldsymbol{\alpha}}$ and its corresponding deconvolution estimate $\boldsymbol{g}(\hat{\boldsymbol{\alpha}})$.

Rather than $l(\boldsymbol{\alpha})$ **deconvolveR** maximizes a penalized log likelihood

$$m(\boldsymbol{\alpha}) = l(\boldsymbol{\alpha}) - s(\boldsymbol{\alpha}), \tag{23}$$

where

$$s(\boldsymbol{\alpha}) = c_0 \left(\sum_{h=1}^{p} \alpha_h^2\right)^{1/2} = c_0 \|\boldsymbol{\alpha}\|; \tag{24}$$

$c_0 = 1$ is the default value in **deconvolveR**. Standard asymptotic calculations give

$$\mathrm{cov}(\boldsymbol{\alpha}) = \{\boldsymbol{\mathcal{I}}(\boldsymbol{\alpha}) + \ddot{\boldsymbol{s}}(\boldsymbol{\alpha})\}^{-1} \boldsymbol{\mathcal{I}}(\boldsymbol{\alpha}) \{\boldsymbol{\mathcal{I}}(\boldsymbol{\alpha}) + \ddot{\boldsymbol{s}}(\boldsymbol{\alpha})\}^{-1} \tag{25}$$

as an approximate covariance matrix of $\hat{\boldsymbol{\alpha}}$ when α is the true value in model (14). The Hessian matrix $\ddot{s}(\boldsymbol{\alpha})$ in Equation 25 is calculated to be

$$\ddot{\boldsymbol{s}}(\boldsymbol{\alpha}) = \frac{c_0}{\|\boldsymbol{\alpha}\|}\left(I - \frac{\boldsymbol{\alpha}\boldsymbol{\alpha}^\top}{\|\boldsymbol{\alpha}\|^2}\right), \tag{26}$$

I the $p \times p$ identity.

Finally, define the $p \times p$ matrix $\boldsymbol{D}(\boldsymbol{\alpha})$ to be

$$\boldsymbol{D}(\boldsymbol{\alpha}) = \mathrm{diag}\{\boldsymbol{g}(\boldsymbol{\alpha})\} - \boldsymbol{g}(\boldsymbol{\alpha})\boldsymbol{g}(\boldsymbol{\alpha})^\top, \tag{27}$$

$\mathrm{diag}\{\boldsymbol{g}(\boldsymbol{\alpha})\}$ denoting the $m \times m$ diagonal matrix with jth diagonal entry $g_j(\boldsymbol{\alpha})$. Then the approximate covariance matrix of $\boldsymbol{g}(\hat{\boldsymbol{\alpha}})$ is

$$\mathrm{cov}[\boldsymbol{g}(\hat{\boldsymbol{\alpha}})] \doteq \boldsymbol{D}(\boldsymbol{\alpha})\boldsymbol{Q}\,\mathrm{cov}(\boldsymbol{\alpha})\boldsymbol{Q}^\top \boldsymbol{D}(\boldsymbol{\alpha}). \tag{28}$$

Larger values of c_0 shrink $\boldsymbol{g}(\hat{\boldsymbol{\alpha}})$ more forcefully toward the flat prior $\boldsymbol{g} = (1/m, 1/m, \ldots, 1/m)$ (if $\mathcal{T}$ is equally spaced). Looking at Equation 25, a measure of the strength of the penalty term compared to the observed data is the ratio of traces $S(\boldsymbol{\alpha})$,

$$S(\boldsymbol{\alpha}) = \frac{\mathrm{tr}[\ddot{s}(\boldsymbol{\alpha})]}{\mathrm{tr}[\boldsymbol{\mathcal{I}}(\boldsymbol{\alpha})]} = \frac{c_0(p-1)}{\|\boldsymbol{\alpha}\|\,\mathrm{tr}[\boldsymbol{\mathcal{I}}(\boldsymbol{\alpha})]}. \tag{29}$$

$S(\hat{\boldsymbol{\alpha}})$ is printed out by **deconvolveR**, allowing adjustment of c_0 for more or less shrinking if so desired. $S(\hat{\boldsymbol{\alpha}})$ was quite small in our examples, supporting $c_0 = 2$ as a conservative choice.

3. The Shakespeare data

Word counts for the entire Shakespearean canon appear in Table 1: 14,376 distinct words were so rare they appeared just once each, 4,343 twice each, 2,292 three times each, with the table continuing on to the five words observed 100 times each throughout the canon. We assume that the ith distinct word, in a hypothetical listing of Shakespeare's complete vocabulary (not just that seen in the canon), appeared X_i times in the canon, X_i following a Poisson distribution with expectation Θ_i,

$$X_i \sim \text{Poisson}(\Theta_i). \tag{30}$$

As in Efron and Thisted (1976) we are interested in the distribution of the unseen parameters Θ_i, but here based on the g-modeling methodology of Section 2.

The support set $\mathcal{T}$ for Θ (8) was taken to be equally spaced on the

$$\lambda = \log(\theta) \tag{31}$$

scale,

$$\lambda = (-4.000, -3.975, -3.950, \ldots, 4.500), \tag{32}$$

with $m = 341$ support points (a denser grid than turned out to be necessary). Modeling Θ on a log scale is useful here because rare words, with very small values of Θ, comprise the bulk of Shakespeare's vocabulary, as Table 1 suggests.

The sample space $\mathcal{X}$ for X (9) was

$$\mathcal{X} = (1, 2, \ldots, 100). \tag{33}$$

(Eight hundred forty-six distinct words appear more than 100 times each in the canon; these are common words such as "and" or "the" that form the bulk of the canon's approximately

	1	2	3	4	5	6	7	8	9	10
0+	14376	4343	2292	1463	1043	837	638	519	430	364
10+	305	259	242	223	187	181	179	130	127	128
20+	104	105	99	112	93	74	83	76	72	63
30+	73	47	56	59	53	45	34	49	45	52
40+	49	41	30	35	37	21	41	30	28	19
50+	25	19	28	27	31	19	19	22	23	14
60+	30	19	21	18	15	10	15	14	11	16
70+	13	12	10	16	18	11	8	15	12	7
80+	13	12	11	8	10	11	7	12	9	8
90+	4	7	6	7	10	10	15	7	7	5

Table 1: Shakespeare's word counts; $14,376$ distinct words appeared once each in the canon, $4,343$ twice each, etc.

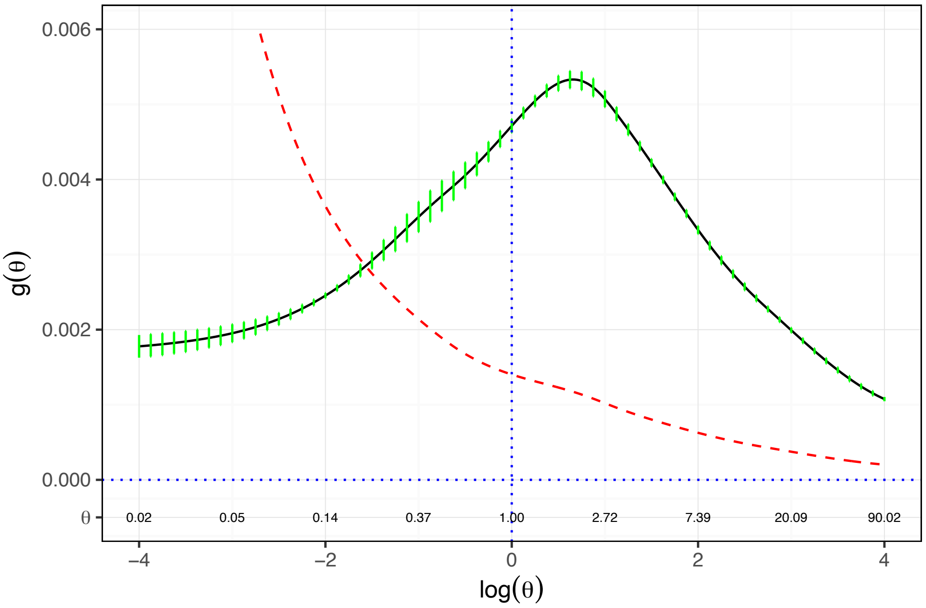

Figure 1: Empirical Bayes deconvolution estimate for Shakespeare word counts. Solid curve is prior $\hat{\boldsymbol{g}} = \boldsymbol{g}(\hat{\boldsymbol{\alpha}})$ is given by Equation 14 and $\hat{\boldsymbol{\alpha}}$ is the maximum likelihood estimate calculated from Equation 21; dashed curve is adjusted prior $\tilde{\boldsymbol{g}}$ in Equation 37 correcting for absent zero counts in Table 1. Vertical green bars are ± one standard error, calculated from diagonal elements of formula (25).

900,000 total count, but they are of less interest here than those at the rarer end of the Θ distribution.) Table 1 gives the count vector $\boldsymbol{y}$, $y_1 = 14,376$, $y_2 = 4,343$, etc.

The structure matrix $\boldsymbol{Q}$ (14) was taken to be a natural spline in λ with five degrees of freedom,

$$\boldsymbol{Q} = \text{ns}(\mathcal{T}, 5) \tag{34}$$

in language R, a 341×5 matrix. Some care is needed in setting the entries p_{kj} of the matrix $\boldsymbol{P}$. Letting

$$\tilde{p}_{kj} = e^{-\theta_{(j)}} \frac{\theta_{(j)}^{x_{(k)}}}{x_{(k)}!}, \tag{35}$$

the entries p_{kj} (10) are

$$p_{kj} = \tilde{p}_{kj} \bigg/ \sum_{h=1}^{100} \tilde{p}_{hj} \,. \tag{36}$$

This compensates for the *truncated data* in Table 1: the zero category – words in Shakespeare's vocabulary he didn't use in the canon – are necessarily missing. (Also missing, less necessarily, are words appearing more than 100 times each.) The definition in Equation 36 makes column j of $\boldsymbol{P}$ into the *truncated* Poisson distribution of X given $\Theta = \theta_{(j)}$. Function `deconv()` in R package **deconvolveR** was run with $\boldsymbol{y}$, $\mathcal{T}$, $\boldsymbol{Q}$, and $\boldsymbol{P}$ as previously specified, and with regularization constant $c_0 = 2$. The solid curve in Figure 1 plots the entries of $\hat{\boldsymbol{g}} = (\cdots \hat{g}_j \cdots)^\top$ (plotted as continuous curve) versus $\lambda_j = \log(\theta_{(j)})$. About 45% of the total mass $\sum \hat{g}_j = 1$ lies below $\Theta = 1$ ($\lambda = 0$), indicating the prevalence of rare words in Shakespeare's usage.

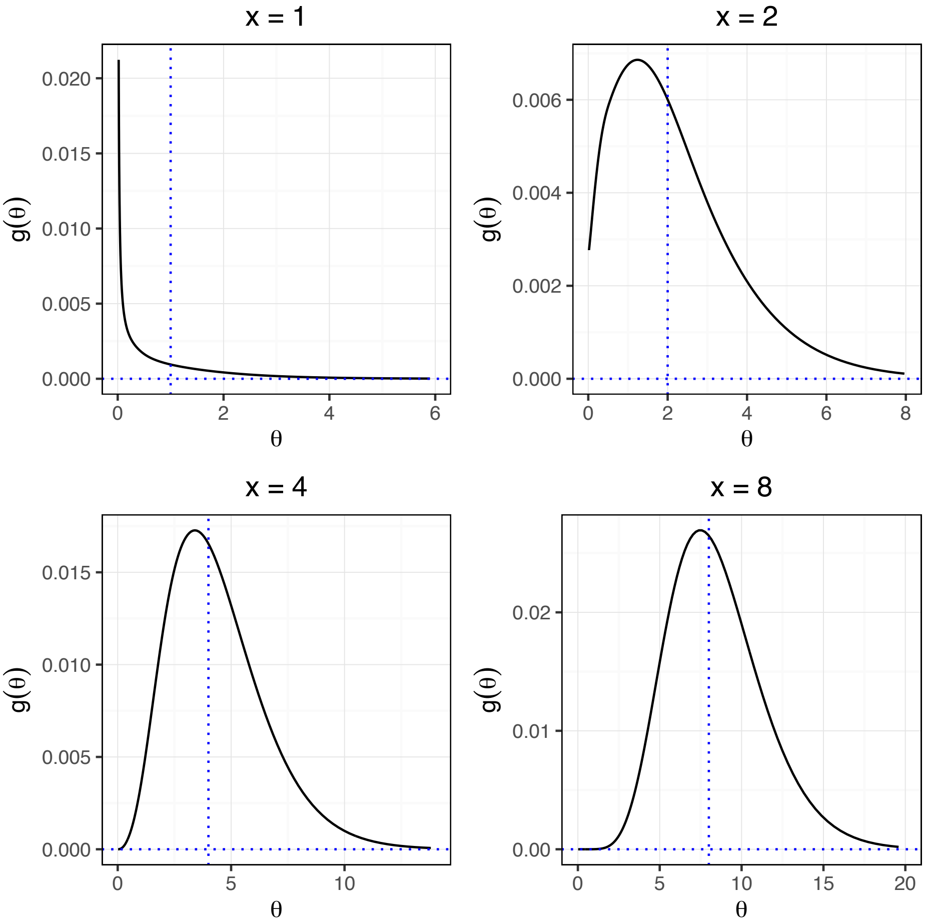

Figure 2: Estimated posterior densities $\tilde{g}(\theta \mid x)$ from Equation 38 for $x = 1, 2, 4, 8$. (Graphs shown actually proportional to $\tilde{g}_j p_{kj}/\theta_{(j)}$ to account for the unequal spacing of $\theta_{(j)}$.) The preponderance of small Θ values pulls the mode of $\tilde{g}(\theta \mid x)$ below x but less so as x increases.

Forty-five percent is an underestimate. A word with parameter Θ_i has probability $\exp(-\Theta_i)$ of yielding $X_i = 0$, in which case it will not be observed. Words with small Θ_i values are systematically thinned out of the observed counts. We can correct for thinning by defining

$$\tilde{g}_j = c_1 \hat{g}_j \Big/ \left(1 - e^{-\theta_{(j)}}\right) , \tag{37}$$

c_1 the constant that makes $\tilde{\boldsymbol{g}}$ sum to 1. The red dashed curve in Figure 1 shows $\tilde{\boldsymbol{g}}$. This is a g-modeling estimate for the prior distribution of Θ we would see if there were no data truncation. It puts 88% of its probability mass below $\Theta = 1$. (See later discussion for some difficulties with this result.) Equation 37 is by no means obvious. It helps to consider a simple case: suppose Θ can take on only two possible values, $\mathcal{T} = \{1, 10\}$, with equal prior probabilities (in the untruncated situation), say $\tilde{g}_1(1) = \tilde{g}_2(10) = 0.5$. If we begin with some large number $\tilde{N}$ of draws $X_i \sim \text{Poisson}(\Theta_i)$, we will observe about $\tilde{N}/2$ from $\Theta = 10$, but only about $0.632\tilde{N}/2$ from $\Theta = 1$ since $X_i = 0$ is unobservable. That is, we will effectively have prior weights

$$g_1(1) = \frac{0.632}{1 + 0.632} = 0.387 \quad \text{and} \quad g_2(10) = \frac{1}{1 + 0.632} = 0.613.$$

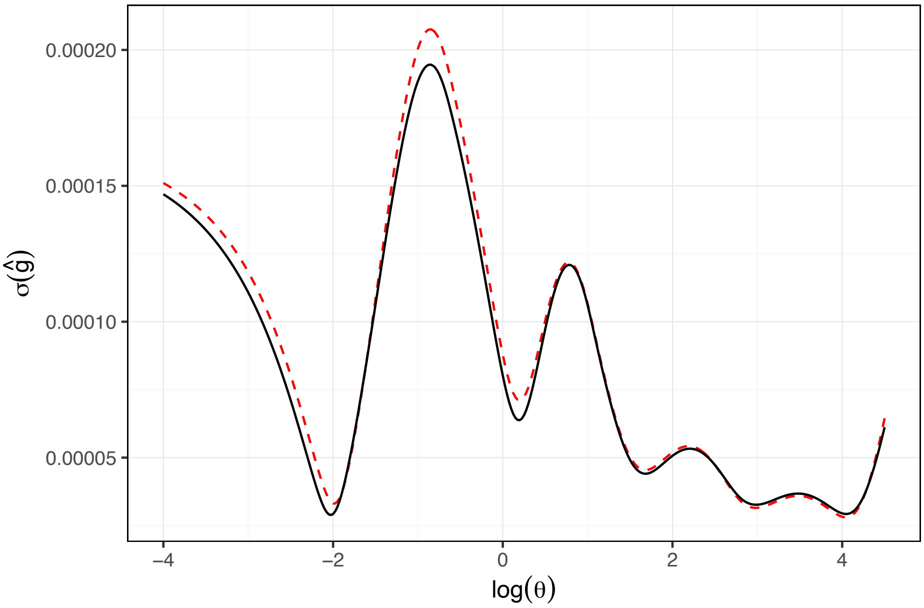

Figure 3: Estimated standard errors for components of $\hat{g}$ in Figure 1. Solid curve from $B = 200$ parametric bootstrap replications (39); dashed curve from theoretical formula (25).

The estimated prior $\tilde{\boldsymbol{g}} = (\tilde{g}_1, \tilde{g}_2, \ldots, \tilde{g}_m)$ can be used to carry out Bayesian computations for the Θ_i parameters, for instance, calculating the posterior probabilities

$$\tilde{g}\left(\Theta_i = \theta_{(j)} \mid X_i = x_{(k)}\right) = c_k \tilde{g}_j p_{kj}, \tag{38}$$

where p_{kj} is the density in Equation 36 and $c_k = (\sum \tilde{g}_j p_{jk})^{-1}$. The cases $x_{(k)}$ equal to 1, 2, 4, and 8 appear in Figure 2, now graphed versus θ instead of $\log(\theta)$. (To compensate for the unequal spacing of the $\theta_{(j)}$ values, the graphs are actually proportional to $\tilde{g}_j p_{kj}/\theta_{(j)}$.) The preponderance of small Θ values seen in Figure 1 pulls the mode of $\tilde{g}(\theta \mid x)$ toward zero, though less so for larger x.

The vertical green bars in Figure 1 indicate ± one standard error for $\hat{g}_j$. These were obtained as the square roots of the diagonal elements of $\text{cov}(\hat{\boldsymbol{g}})$ from Equation 28. As a check on the obtained values, a parametric bootstrap simulation was run: bootstrap count vectors

$$\boldsymbol{y}^\star \sim \text{mult}_n\left(N, \hat{\boldsymbol{f}}\right) \tag{39}$$

($n = 100$, the length of $\boldsymbol{x}_0$, and $N = 30,688$, the total number of counts in Table 1) were obtained, with the maximum likelihood estimation $\hat{\boldsymbol{f}} = \boldsymbol{f}(\hat{\boldsymbol{\alpha}})$ replacing $\boldsymbol{f}$ in Equation 13; then $\hat{\boldsymbol{\alpha}}^\star$ was computed as the maximizer of $\sum y_k^\star \log f_k(\boldsymbol{\alpha})$ in Equation 17, giving $\boldsymbol{g}(\hat{\boldsymbol{\alpha}}^\star)$ as in Equation 14. Finally bootstrap standard errors for the components of $\hat{\boldsymbol{g}}$ were calculated from $B = 200$ simulations. Figure 3 compares the theoretical standard errors from Equation 25 with their bootstrap counterparts. The agreement is quite good in this case. In practice the bootstrap calculations are usually easy to carry out as a reassuring supplement to the theory.

Looking back at Table 1, it is tempting to ask how many "new" words (i.e., distinct words not appearing in the canon) we might find in a trove of newly discovered Shakespeare. This is Fisher's famous *missing species problem*.

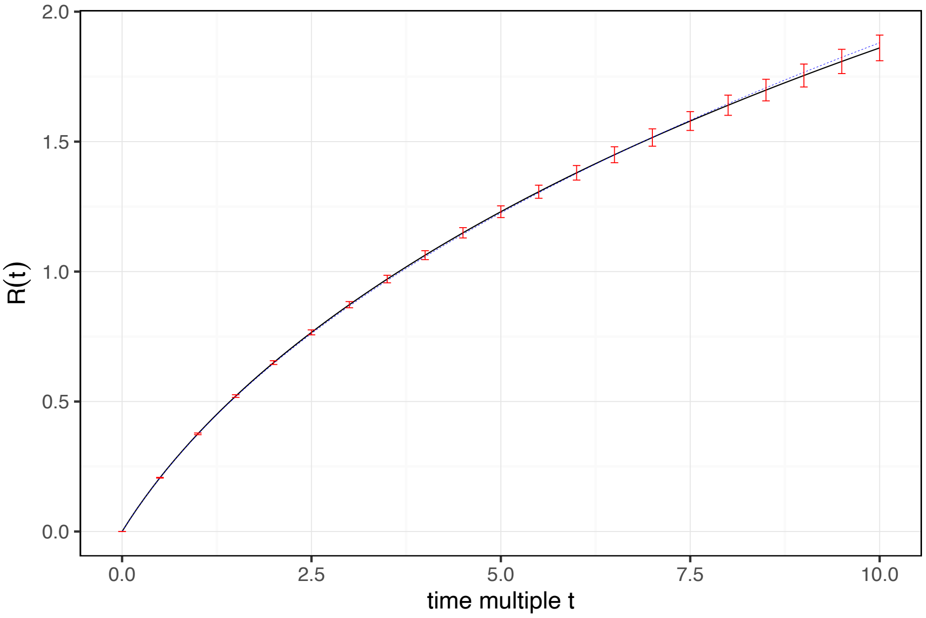

Figure 4: Predicted ratio of distinct new words found in t newly discovered Shakespeare canons, relative to the observed number $N = 30,688$ already seen. Bars indicate ± one standard error, as derived from (28) and (41). Light dashed line shows predictions from Fisher's gamma model (43)–(44).

Suppose then that a previously unknown Shakespearean corpus of length $t \cdot C$ were found, $C \doteq 900,000$ the length of the known canon. Assuming a Poisson process model with intensity Θ_i for word i, the probability that word i did not appear in the canon but does appear in the new corpus is

$$e^{-\Theta_i}\left(1 - e^{-\Theta_i t}\right); \tag{40}$$

Equation 40 and definition (37) give, after some work, an estimate for $R(t)$, the expected number of distinct new words found, divided by N, the observed number of distinct words in the canon:

$$R(t) = \sum_{j=1}^{m} \hat{g}_j r_j(t), \tag{41}$$

$$r_j = \frac{e^{-\theta_{(j)}}}{1 - e^{-\theta_{(j)}}}\left(1 - e^{-\theta_{(j)} t}\right). \tag{42}$$

A graph of Shakespeare's $R(t)$ function is shown in Figure 4, along with standard error bars derived from Equation 25. It predicts $R(t) = 1$, that is a doubling of Shakespeare's observed vocabulary, at $t = 3.74$.

All of this might seem like the rankest kind of statistical speculation. In fact though, formula (41) performs well in cross-validatory tests where part of the canon is set aside and then predicted from the remainder. See Thisted and Efron (1987).

Fisher's original proposal for the missing species problem took the prior $g(\theta)$ to be an (improper) gamma density,

$$g(\theta) = c\theta^{\gamma-1}e^{-\theta/\beta}. \tag{43}$$

This is of form (14), now with prior degrees of freedom p just 2. Applied to Shakespeare's word counts, Equation 43 gave maximum likelihood estimates

$$\hat{\gamma} = -0.3954, \qquad \hat{\beta} = 104.263. \tag{44}$$

The resulting prediction curve, shown in Figure 4, is nearly the same as that for our five degrees of freedom spline model.

The missing species problem has an inestimable aspect at its rarest extreme: if Shakespeare knew 1,000,000 words that he only employed once each in a million canons, these would remain effectively invisible to us. By taking $\theta_{(1)} = \exp(-4) = 0.018$ at (32), our model legislates out the one-in-a-million cases. It gives a good fit to the data, with

$$\hat{\boldsymbol{y}} = N \cdot \boldsymbol{P}\hat{\boldsymbol{g}} \tag{45}$$

passing a Wilks' test for fit to the observed count vector $\boldsymbol{y}$ – so in this sense it cannot be improved by lowering $\theta_{(1)}$.

All of this seems mainly of pedantic interest in the Shakespeare example; less so, however, in biological applications of the missing species problem, where, for instance, the occurrence rates of cloned DNA segments can range over many orders of magnitude.

4. A guide to a new package deconvolveR

The R package **deconvolveR** contains one main function `deconv()` that handles three exponential families, *Binomial*, *Normal* and *Poisson* directly. Since users may wish to experiment with other exponential family models or change the details of how $\boldsymbol{Q}$ is normalized, `deconv()` also accepts user-specified $\boldsymbol{Q}$ and $\boldsymbol{P}$ matrices in its invocation.

The maximum likelihood estimation is carried out using the non-linear optimization function `nlm()` in R with the gradient of the likelihood computed via the theoretical formula in Equation 20. This has a practical effect in that harmless warnings may be generated during the optimization. The Hessian, although available, is not used to guide the optimization in the current version of the software due to numerical considerations.

The package contains a vignette that provides the complete code to reproduce all the results in this paper with additional details. Below we illustrate its use with examples from the three main models, first with simulated data for the Poisson and Normal cases, followed by real data examples using the Shakespeare word count data and an intestinal surgery dataset. We note that although g has discrete support, we plot it as a continuous curve throughout.

4.1. A Poisson simulation

Suppose the Θ_i are drawn from a χ^2 density with 10 degrees of freedom and the $X_i|\Theta_i$ are Poisson with expectation Θ_i :

$$\Theta_i \sim \chi^2_{10} \text{ and } X_i|\Theta_i \sim \text{Poisson}(\Theta_i). \tag{46}$$

We carry out 1000 simulations each with $N = 1000$ observations by first generating the Θ and then creating a 1000×1000 data matrix.

θ	$g(\theta)$	Mean	Standard deviation	Bias	Coefficient of variation
5	5.62	5.44	0.36	-0.12	0.07
10	9.16	9.53	0.49	0.26	0.05
15	4.09	3.34	0.31	-0.07	0.09
20	1.10	0.98	0.22	-0.12	0.23
25	0.23	0.15	0.07	0.06	0.45

Table 2: Simulation results for the Poisson model where the $\Theta_i \sim \chi^2_{10}$ and $X_i|\Theta_i$ are drawn from Poisson(Θ_i) for $i = 1, 2, \ldots, 1000$. The middle 4 columns have been multiplied by 100.

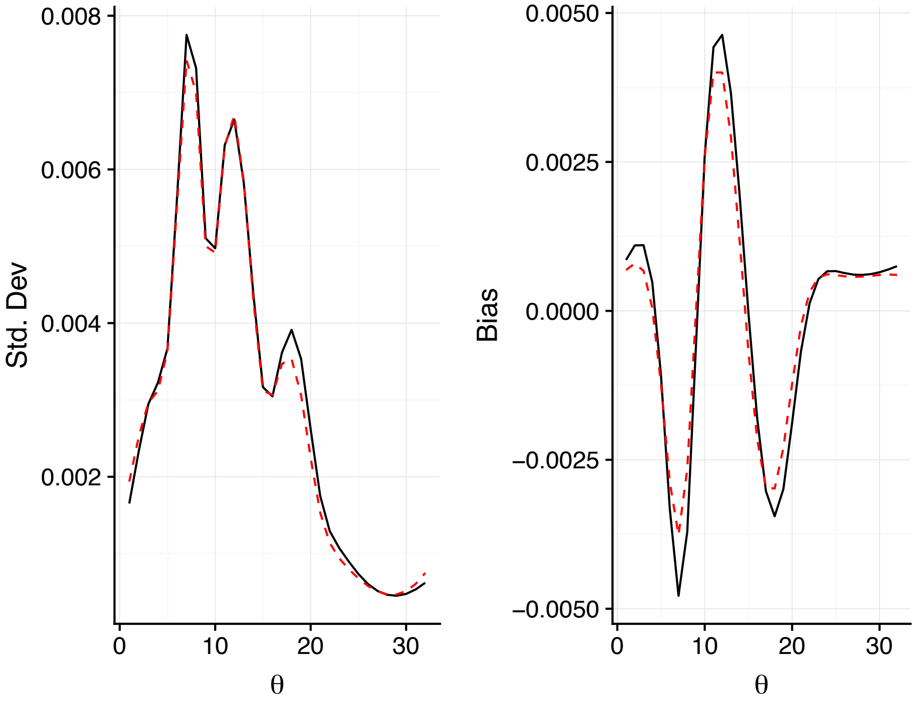

Figure 5: Standard deviations and biases for the simulated Poisson example. Solid curves are from the formulas and the dashed curves are from simulation.

```
R> set.seed(238923)
R> N <- 1000
R> nSIM <- 1000
R> theta <- rchisq(N,  df = 10)
R> data <- sapply(seq_len(nSIM), function(x) rpois(n = N, lambda = theta))
```

Taking the support of Θ to be the discrete set $\mathcal{T} = (1, 2, \ldots, 32)$, we apply the `deconv()` function on each column of the matrix to obtain the estimate $\hat{g}$ along with a host of other statistics.

```
R> tau <- seq(1, 32)
R> results <- apply(data, 2,
+    function(x) deconv(tau = tau, X = x, ignoreZero = FALSE))
```

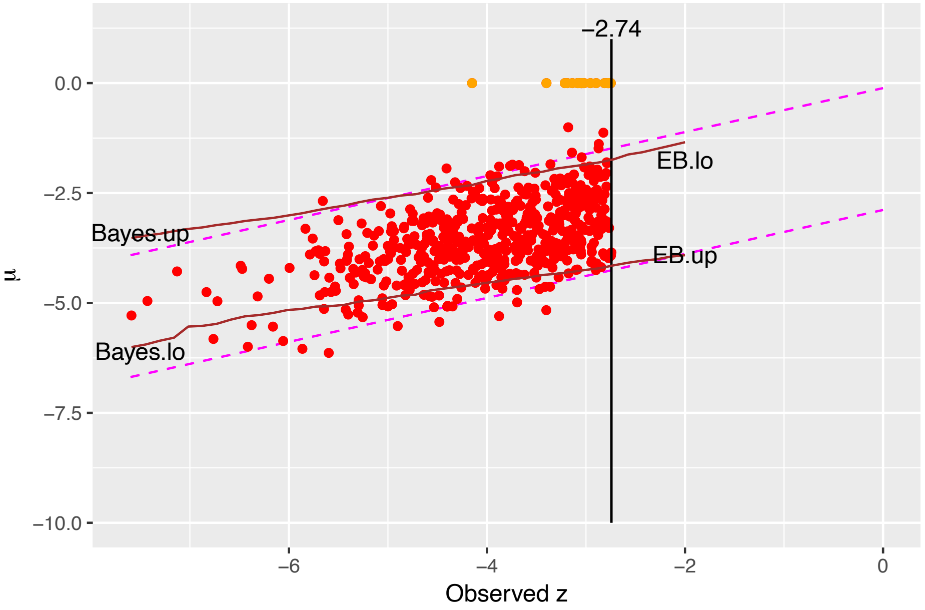

Figure 6: Estimates from g-modeling for a Normal model along with 95% credible intervals (EB.up and EB.lo). They are a close match to the actual Bayes intervals.

Note the use of the `ignoreZero` above – here, unlike the Shakespeare example zeros are observed.

We have once again relied on the default *Poisson* family and a natural cubic spline basis of degree 5 for $\boldsymbol{Q}$. The columns of $\boldsymbol{Q}$ are standardized to have mean zero and sum of squares 1. The regularization (`c0`) parameter is left at the default value of 1. We construct a table for $\hat{g}(\theta)$ and related statistics.

```
R> stats <- sapply(results, function(x) x$stats$mat[, "g"])
R> mean <- apply(stats, 1, mean)
R> sd <- apply(stats, 1, sd)
R> gTheta <- pchisq(tau, df = 10) - pchisq(c(0, tau[-length(tau)]), df = 10)
R> table1 <- data.frame(theta = tau, gTheta = 100 * gTheta,
+    Mean = 100 * mean, StdDev = 100 * sd, Bias = 100 * bias)
```

Table 2 shows that the $g(\theta)$ estimates are reasonable although the coefficient of variation grows larger for values of θ in the tails.

Figure 5 compares the empirical standard deviations (obtained from square roots of the diagonal elements of $\mathrm{cov}[\boldsymbol{g}(\hat{\boldsymbol{\alpha}})]$ in Equation 28) and biases of $g(\hat{\alpha})$ (obtained using Equation 34 of Efron 2016) for a few chosen values of Θ, which perform well here.

4.2. A Normal model

Next, we consider data generated as follows:

$$z_i \sim N(\mu_i, 1) \text{ where } \mu_i = \begin{cases} 0, & \text{with probability } .9 \\ N(-3,1), & \text{with probability } .1 \end{cases} \text{ for } i = 1, 2, \ldots, 10000. \qquad (47)$$

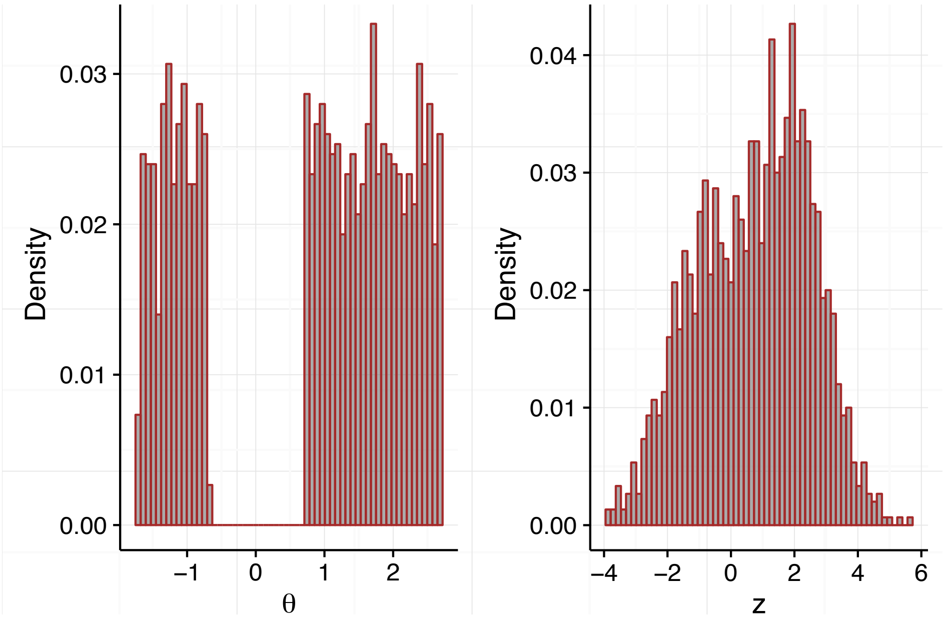

Figure 7: A bimodal distribution for θ on the left and a histogram of $Z_i \sim N(\theta_i, 1)$ on the right.

To deconvolve this data, we specify an atom at zero using the parameter `deltaAt` which applies only to the Normal case. Using $\tau = (-6, -5.75, \dots, 3)$ and a fifth-degree polynomial for the Q basis yields an estimated probability for $\mu = 0$ as 0.887 ± 0.009 with a bias of about -0.006.

```
R> tau <- seq(from = -6, to = 3, by = 0.25)
R> result <- deconv(tau = tau, X = data$z, deltaAt = 0, family = "Normal")
```

The density estimates removing the atom at zero are not accurate at all, but the g-modeling estimates of conditional 95% credible intervals (code included in the package vignette) for μ given z are a good match for the Bayes intervals as shown in Figure 6.

4.3. A twin towers model

In the previous example, the distribution of θ had a significant atom at 0 and the rest of the density was smeared around -3. We now consider the case where θ has a bimodal distribution (included in the package as the dataset `disjointTheta`). Figure 7 is a histogram of the the θ alongside a histogram of the data, generated using $Z_i \sim N(\theta_i, 1)$. Figure 8, on the left, reveals the effect of varying the degrees of freedom of the natural spline basis and regularization parameter. In this case, a choice of 6 or 7 appears reasonable to capture the bimodality. The right side of Figure 8 shows the effect of the regularization parameter c_0 on the estimates: larger values for c_0 smooth out the $\hat{\boldsymbol{g}}$ making the bimodality less prominent. Here $S(\hat{\boldsymbol{\alpha}})$ can serve as a guide for choosing c_0. For varying degrees of freedom, Table 3 shows the estimates of $S(\hat{\boldsymbol{\alpha}})$. A choice of $c_0 \leq 4$ would avoid excessive penalization.

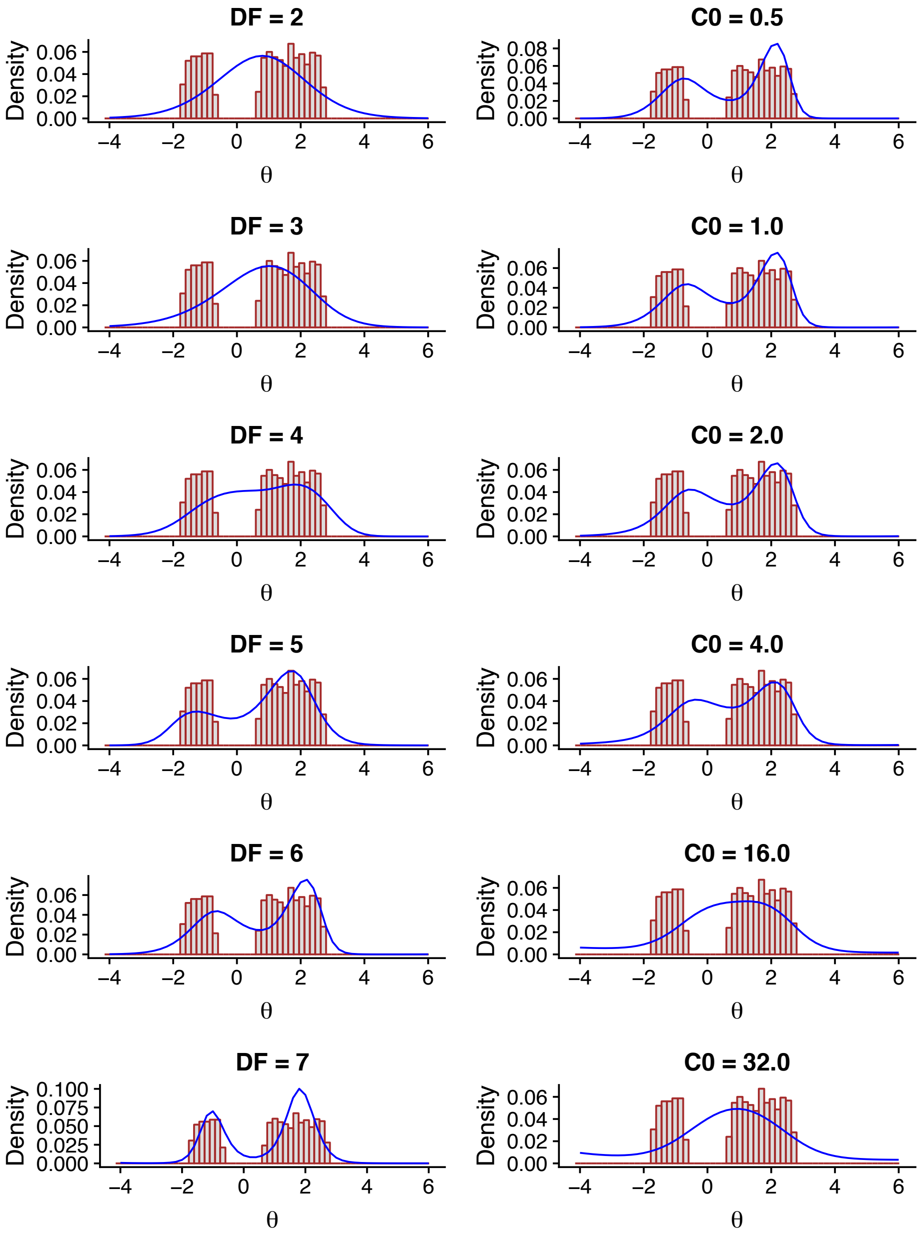

Figure 8: Estimates $\hat{\boldsymbol{g}}$ obtained by varying the degrees of freedom for the natural spline basis and the regularization parameter c_0. On the left, the degrees of freedom are varied for $c_0 = 1$ suggesting a value of 6 or 7. On the right the penalization parameter c_0 is varied for 7 degrees of freedom. Larger values of c_0 smooth out the bimodality as do smaller degrees of freedom.

4.4. Shakespeare example

The data for the Shakespeare example is included in the package as dataset `bardWordCount`. Here, the data is a (truncated) vector of Poisson counts for frequencies of words that appeared exactly once, twice, etc. all the way to 100. We construct the support set $\mathcal{T}$, equally spaced on the $\lambda = \log(\theta)$ scale and call `deconv()` as shown below.

DF	$c_0 = 0.5$	$c_0 = 1$	$c_0 = 2.0$	$c_0 = 4.0$	$c_0 = 8.0$	$c_0 = 16.0$	$c_0 = 32.0$
5	0.00	0.00	0.01	0.02	0.07	0.19	0.48
6	0.00	0.00	0.01	0.02	0.07	0.20	0.54
7	0.00	0.00	0.01	0.03	0.07	0.20	0.57

Table 3: Estimates of $S(\hat{\boldsymbol{\alpha}})$ for the twin towers example from Equation 29 for various values of c_0 using 5, 6, 7 degrees of freedom (column DF) for the natural spline basis.

```
R> lambda <- seq(-4, 4.5, .025)
R> tau <- exp(lambda)
R> result <- deconv(tau = tau, y = bardWordCount, n = 100, c0 = 2)
R> stats <- result$stats
R> head(stats)
```

```
      theta       g    SE.g       G    SE.G  Bias.g     tg
[1,] 0.0183 0.00178 0.000151 0.00178 0.000151 0.000142 0.0184
[2,] 0.0188 0.00178 0.000151 0.00356 0.000302 0.000142 0.0180
[3,] 0.0193 0.00178 0.000150 0.00534 0.000452 0.000141 0.0176
[4,] 0.0197 0.00179 0.000150 0.00713 0.000601 0.000141 0.0172
[5,] 0.0202 0.00179 0.000149 0.00892 0.000751 0.000140 0.0168
[6,] 0.0208 0.00179 0.000149 0.01071 0.000899 0.000140 0.0164
```

```
R> tail(stats)
```

```
        theta        g     SE.g     G     SE.G  Bias.g       tg
[336,]   79.4 0.000923 4.75e-05 0.995 2.87e-04 5.20e-06 0.000174
[337,]   81.5 0.000916 5.06e-05 0.996 2.36e-04 4.85e-06 0.000172
[338,]   83.5 0.000910 5.38e-05 0.997 1.82e-04 4.48e-06 0.000171
[339,]   85.6 0.000903 5.73e-05 0.998 1.25e-04 4.11e-06 0.000170
[340,]   87.8 0.000897 6.08e-05 0.999 6.45e-05 3.73e-06 0.000169
[341,]   90.0 0.000891 6.45e-05 1.000 6.49e-11 3.34e-06 0.000168
```

By default, `deconv()` assumes a Poisson family and works on a sample at a time. The invocation above provided the prior support `tau` $= \mathcal{T}$, the sufficient statistic of counts $\boldsymbol{y}$ and indicated the support of X via the $n = 100$ parameter so that $\mathcal{X} = (1, 2, \ldots, 100)$. The parameter `c0` is the regularization parameter in Equation 24.

The result is a list with a number of quantities, including the MLE $\hat{\boldsymbol{\alpha}}$, the covariance matrix of $\hat{\boldsymbol{\alpha}}$, the matrices $\boldsymbol{P}$ and $\boldsymbol{Q}$ etc. Above, we print the head and tail rows of the `stats` component that includes $\hat{\boldsymbol{g}}$, cumulative $\hat{\boldsymbol{G}}$, standard errors and biases. But one could also print out the ratio of traces $S(\alpha)$ of Equation 29 for example.

```
R> print(result$S)
```

```
[1] 0.005534954
```

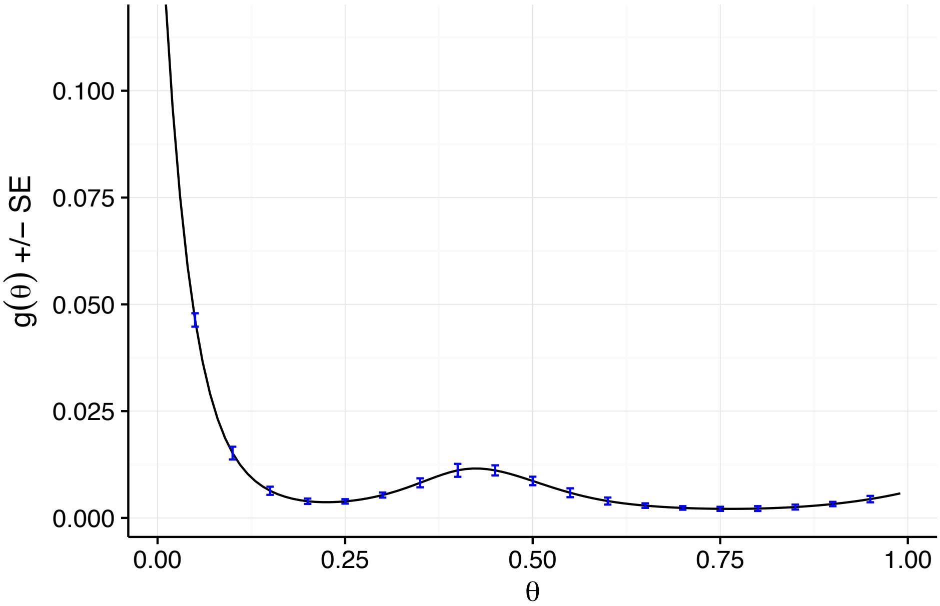

Figure 9: Estimates from g-modeling for a Normal model along with 95% credible intervals. They are a close match to the actual Bayes intervals.

θ	$\hat{g}(\theta)$	SE_f	SE_s	Bias_f	Bias_s
0.01	12.326	0.870	0.911	−0.482	−0.543
0.12	1.033	0.127	0.135	0.051	0.058
0.23	0.369	0.054	0.061	0.023	0.027
0.34	0.757	0.093	0.091	−0.007	−0.008
0.45	1.113	0.118	0.113	−0.037	−0.036
0.56	0.543	0.102	0.097	0.015	0.016
0.67	0.262	0.046	0.049	0.021	0.024
0.78	0.213	0.053	0.050	0.018	0.018
0.89	0.308	0.052	0.046	0.014	0.013
0.99	0.575	0.158	0.157	−0.010	−0.014

Table 4: Comparison of theoretical and bootstrap estimates of standard error (SE) for the surgery data using a Binomial model. The subscripts f and s denote formula and simulation values. All values except the first column have been multiplied by 100.

This indicates that the penalty term $c_0 = 2$ used above is not too big compared to the observed data.

4.5. Intestinal surgery example

The dataset `surg` contains data on intestinal surgery on 844 cancer patients. In the study, surgeons removed *satellite* nodes for later testing. The data consists of pairs (n_i, X_i) where n_i is the number of satellites removed and X_i is the number found to be malignant among them. We assume a Binomial model with $X_i \sim \mathit{Binomial}(n_i, \theta_i)$ with θ_i being the probability of any one satellite site being malignant for the ith patient.

We take $\tau = (0.01, 0.02, \ldots, 0.09)$ (so $m = 99$), $\boldsymbol{Q}$ to be a 5-degree natural spline with columns standardized to mean 0 and sum of squares equal to 1 and the penalization parameter at the default value 1.

```
R> tau <- seq(from = 0.01, to = 0.99, by = 0.01)
R> result <- deconv(tau = tau, X = surg, family = "Binomial")
```

Figure 9 shows the estimated prior density $\hat{g}(\theta)$ with error bars one standard error above and below. The figure shows a large node near $\Theta = 0$ with a 50% chance of $\Theta \leq 0.09$ and the remaining 50% spread out almost evenly over $[0.1, 1.0]$.

As a check on the estimates of standard error and bias provided by `deconv()`, we compare the results with what we obtain using a parametric bootstrap. The bootstrap is run as follows.

For each of 1000 runs, 844 simulated realizations $\hat{\Theta}^\star$ are sampled from the density $\hat{g}$. Each gave an $X_i \sim \mathit{Binomial}(n_i, \hat{\Theta}^\star)$ with n_i the ith sample in the original dataset. Finally, $\hat{\boldsymbol{\alpha}}^\star$ was computed using `deconv()`. The results are shown in Table 4.

5. Summary

Empirical Bayes estimation exploded into the statistics field around the 1950s as a new branch of statistical inference. Practical tools have been mostly related to f-modeling where probability models are proposed for the marginal density $f(x)$ of the data. Empirical Bayes deconvolution, the problem of estimating the prior g from the data, is harder, with slow nonparametric rates of convergence (Carroll and Hall 1988). Parametric modeling of g (Efron 2016) from a known exponential family offers a way forward and our package **deconvolveR** implements this approach. The examples shown here indicate that it works well for a range of problems.

Acknowledgments

Balasubramanian Narasimhan's work supported in part by the Clinical and Translational Science Award 1UL1 RR025744 for the Stanford Center for Clinical and Translational Education and Research (Spectrum) from the National Center for Research Resources, National Institutes of Health and award LM07033 from the National Institutes of Health. Bradley Efron's work supported by NSF award DMS 1608182.

References

Carroll RJ, Hall P (1988). "Optimal Rates of Convergence for Deconvolving a Density." *Journal of the American Statistical Association*, **83**(404), 1184–1186. doi:10.1080/01621459.1988.10478718.

Efron B (2010). *Large-Scale Inference: Empirical Bayes Methods for Estimation, Testing, and Prediction*, volume 1 of *Institute of Mathematical Statistics Monographs*. Cambridge University Press, Cambridge.

Efron B (2014). "Two Modeling Strategies for Empirical Bayes Estimation." *Statistical Science*, **29**(2), 285–301. doi:10.1214/13-sts455.

Efron B (2016). "Empirical Bayes Deconvolution Estimates." *Biometrika*, **103**(1), 1–20. doi:10.1093/biomet/asv068.

Efron B, Narasimhan B (2020). ***deconvolveR**: Empirical Bayes Estimation Strategies.* R package version 1.2.1, URL https://CRAN.R-project.org/package=deconvolveR.

Efron B, Thisted R (1976). "Estimating the Number of Unseen Species: How Many Words Did Shakespeare Know?" *Biometrika*, **63**(3), 435–447.

Gentleman RC, Carey VJ, Bates DM, Bolstad B, Dettling M, Dudoit S, Ellis B, Gautier L, Ge Y, Gentry J, Hornik K, Hothorn T, Huber W, Iacus S, Irizarry R, Leisch F, Li C, Maechler M, Rossini AJ, Sawitzki G, Smith C, Smyth G, Tierney L, Yang JYH, Zhang J (2004). "**Bioconductor**: Open Software Development for Computational Biology and Bioinformatics." *Genome Biology*, **5**(10), R80. URL http://genomebiology.com/2004/5/10/R80.

Hardcastle TJ (2019). ***baySeq**: Empirical Bayesian Analysis of Patterns of Differential Expression in Count Data.* R package version 2.20.0, URL https://bioconductor.org/packages/baySeq/.

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Robinson MD, McCarthy DJ, Smyth GK (2010). "**edgeR**: A **Bioconductor** Package for Differential Expression Analysis of Digital Gene Expression Data." *Bioinformatics*, **26**(1), 139–140. doi:10.1093/bioinformatics/btp616.

Schwender H (2020). ***siggenes**: Multiple Testing Using SAM and Efron's Empirical Bayes Approaches.* R package version 1.62.0, URL https://bioconductor.org/packages/siggenes/.

Thisted R, Efron B (1987). "Did Shakespeare Write a Newly-Discovered Poem?" *Biometrika*, **74**(3), 445–455.

Affiliation:

Balasubramanian Narasimhan, Bradley Efron
Department of Statistics
Sequoia Hall
Stanford University
390 Serra Mall, United States of America
E-mail: naras@stanford.edu, brad@stat.stanford.edu
URL: https://web.stanford.edu/~naras/, http://efron.web.stanford.edu/

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
August 2020, Volume 94, Issue 11 *Submitted:* 2017-11-30
doi:10.18637/jss.v094.i11 *Accepted:* 2019-04-02