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 (unobservable) $\Theta_1, \Theta_2, \ldots, \Theta_N$, 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 $N = 6033$ genes, with X_i measuring a standardized difference between patients and controls for the ith gene.

A Bayesian analysis of situation (1) 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 case (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 (2). Both $\hat{g}(\theta)$ and $\hat{g}(\theta \mid x)$ can be of interest in applied problems.

In the normal model (1), $f(x)$ 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. 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 (1). We might, for example, have

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

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 estimationg $g(\theta)$ in (3) given a random sample from $f(x)$.

Section 2 presents a brief review of g-modeling estimation theory, illustrated in Section 3 with a Poisson example (4) 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 2014) package **deconvolveR**, for the empirical Bayes estimation of $g(\theta)$ and $g(\theta \mid x)$.

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 $\boldsymbol{X} = (X_1, X_2, \ldots, X_N)$.

For computational convenience we assume that Θ's sample space $\mathcal{T}$ is finite and discrete,

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

The $\theta_{(j)}$ values need not be equally spaced, and in fact are not in the Shakespeare example of Section 3. 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 (1), the X_i can be discretized by binning.)

In the discrete formulation (8), the prior $g(\theta)$ is represented by a vector $\boldsymbol{g} = (g_1, g_2, \ldots, g_m)^\top$. Likewise, the marginal $f(x)$ (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

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

we define the $n \times m$ matrix $\boldsymbol{P} = (p_{kj})$. Now the convolution-type relationship (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 $\boldsymbol{g}$,

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

G-modeling assumes that $g(\theta)$ is a member of a p-parameter exponential family on $\mathcal{T}$, expressed in the discrete formulation (8) as

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

where $\boldsymbol{Q}$ is an $m \times p$ *structure matrix*, the default choice in **deconvolveR** being the natural spline basis $\text{ns}(\mathcal{T}, p)$; α is the unknown p-dimensional natural parameter vector; $c(\alpha)$ is the divisor necessary to make $\boldsymbol{g}$ sum to 1. Coordinate-wise, (14) says that

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

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

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

The log likelihood function $l(\alpha)$ of $\boldsymbol{y}$ in model (11)–(14) is

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

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

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

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

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

for $k = 1, 2, \ldots, n$.

The *score function* for α then turns out to be

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

where

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

The maximum likelihood estimate $\hat{\alpha}$ is found by numerically maximizing $l(\alpha)$ (17) or by solving $\dot{l}(\hat{\alpha}) = 0$ (20).

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

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

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

Rather than $l(\alpha)$ (17) **deconvolveR** maximizes a penalized log likelihood

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

where

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

$c_0 = 2$ is the default value in **deconvolveR**. Standard asymptotic calculations give

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

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

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

I the $p \times p$ identity.

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

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

	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, 4343 twice each, etc.

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

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

Larger values of c_0 in (23)–(24) shrink $\boldsymbol{g}(\hat{\alpha})$ more forcefully toward the flat prior $\boldsymbol{g} = {}^{(1,1,\ldots,1)}/_{m}$. Looking at (25), a measure of the strength of the penalty term compared to the observed data is the ratio of traces $S(\alpha)$,

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

$S(\hat{\alpha})$ is printed out by **deconvolveR**, allowing adjustment of c_0 for more or less shrinking if so desired.

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, 4343 twice each, 2292 three times each, 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 entire vocabulary, appeared X_i times in the canon, X_i following a Poisson distribution with expectation Θ_i,

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

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. 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 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 $\boldsymbol{y}$.

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

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

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

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

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.) Definition (36) makes column j of $\boldsymbol{P}$ into the *truncated* Poisson distribution of X given $\Theta = \theta_{(j)}$.

Program **deconvolveR** was run with $\boldsymbol{y}$, $\mathcal{T}$, $\boldsymbol{Q}_0$, and $\boldsymbol{P}_0$ as previously specified, and with regularization constant $c_0 = 2$ (24). The solid curve in Figure 1 plots the entries of $\hat{\boldsymbol{g}} = (\cdots \hat{g}_j \cdots)^\top$ versus $\lambda_j = \log(\theta_{(j)})$. About 45% of the total mass $\sum \hat{g}_j = 1$ lies below $\Theta = 1$, indicating the prevalence of rare words in Shakespeare's usage.

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.)

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

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

where $\tilde{p}_{kj}$ is the Poisson density (35) and $c_k = 1/\sum \tilde{g}_j \tilde{p}_{jk}$. 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

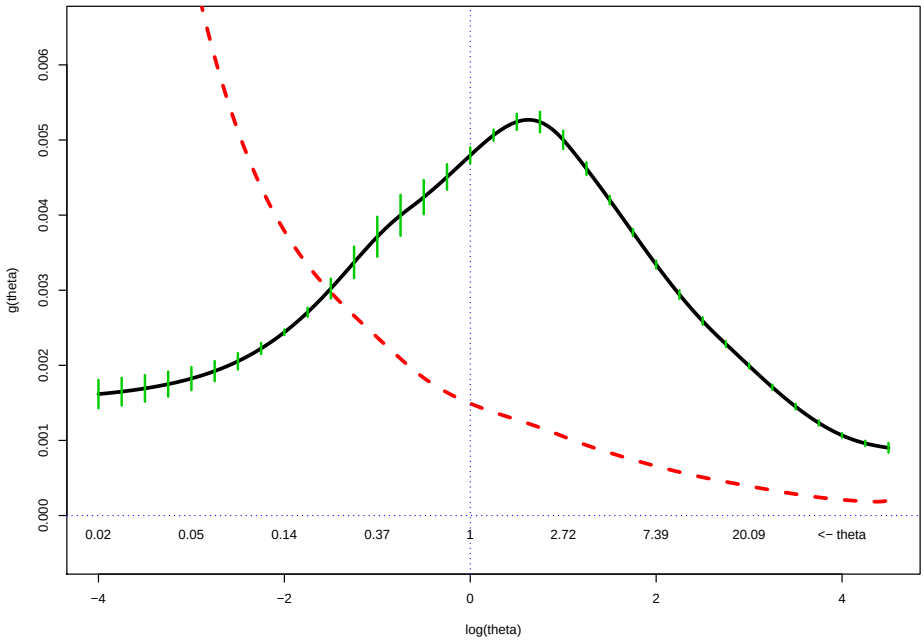


Figure 1: Empirical Bayes deconvolution estimate for Shakespeare word counts. Solid curve is prior $\hat{g}$ in model (30); dashed curve is adjusted prior $\tilde{\boldsymbol{g}}$ (37) correcting for absent zero counts in Table 1. Vertical green bars are $\pm$ one standard error, calculated from diagonal elements of formula (25).

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 $\pm$ one standard error for $\hat{g}_j$. These were obtained as the square roots of the diagonal elements of $\operatorname{cov} \hat{\boldsymbol{g}}$ (28). As a check on (28), a parametric bootstrap simulation was run: bootstrap count vectors

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

were obtained, with the MLE $\hat{\boldsymbol{f}} = \boldsymbol{f}(\hat{\alpha})$ replacing $\boldsymbol{f}$ in (13); then $\hat{\alpha}^*$ was computed as the maximizer of $\sum y_k^* \log f_k(\alpha)$ (17), giving $\boldsymbol{g}(\hat{\alpha}^*)$ as in (14). Finally bootstrap standard errors for the components of $\hat{\boldsymbol{g}}$ were calculated from $B = 200$ simulations of (39).

Figure 3 compares the theoretical standard errors from (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*.

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

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

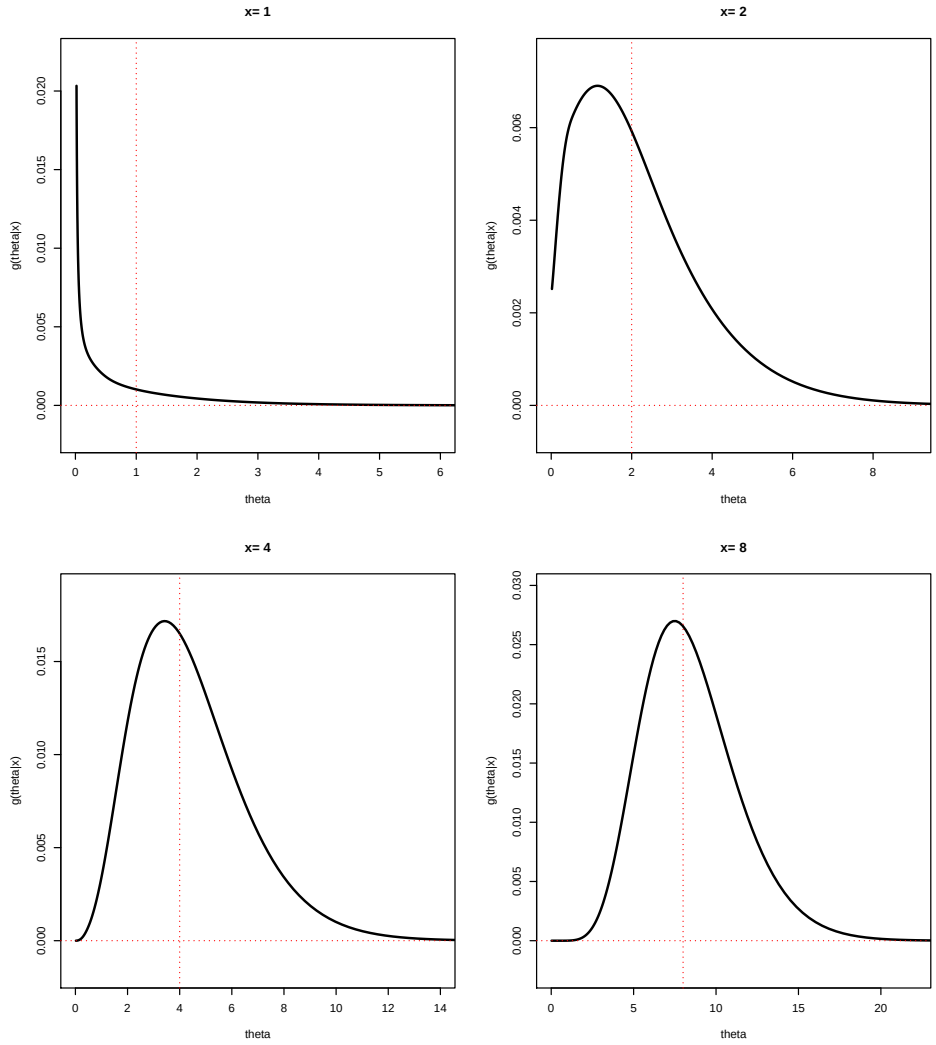


Figure 2: Estimated posterior densities $\tilde{g}(\theta \mid x)$ (38) for $x = 1, 2, 4, 8$. The preponderance of small Θ values pulls the mode of $\tilde{g}(\theta \mid x)$ below x but less so as x increases.

(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 (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

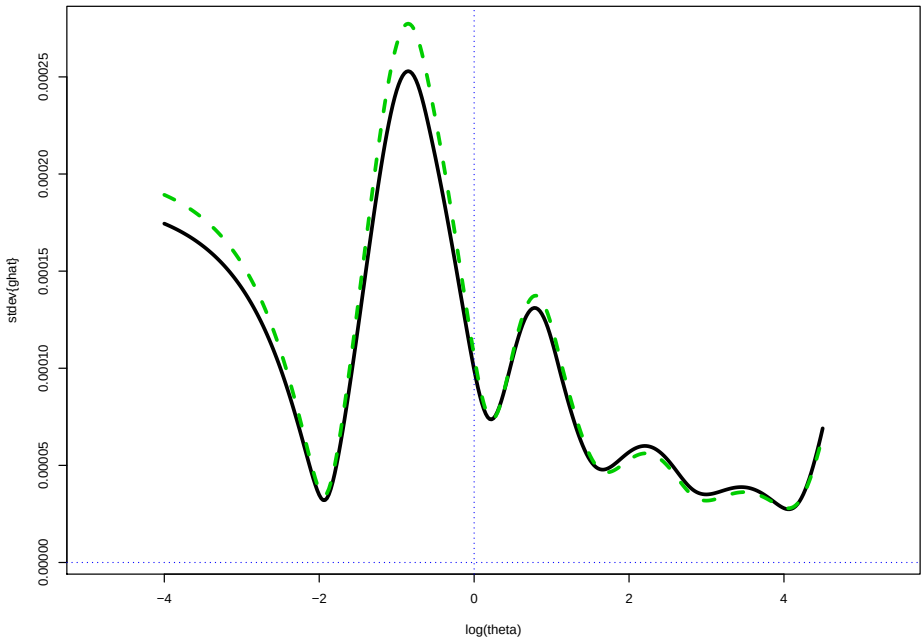


Figure 3: Estimated standard errors for components of $\hat{g}$ in Figure 1. Solid curve from $B = 200$ parametric bootstrap replications (39); dashed curve from theoretical formula (25).

predicted from the remainder. See Thisted and Efron (1987).

Fisher's original proposal for the missing species problem took the prior $g(\theta)$ (5) to be an (improper) gamma density,

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

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

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

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 pedantic 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.

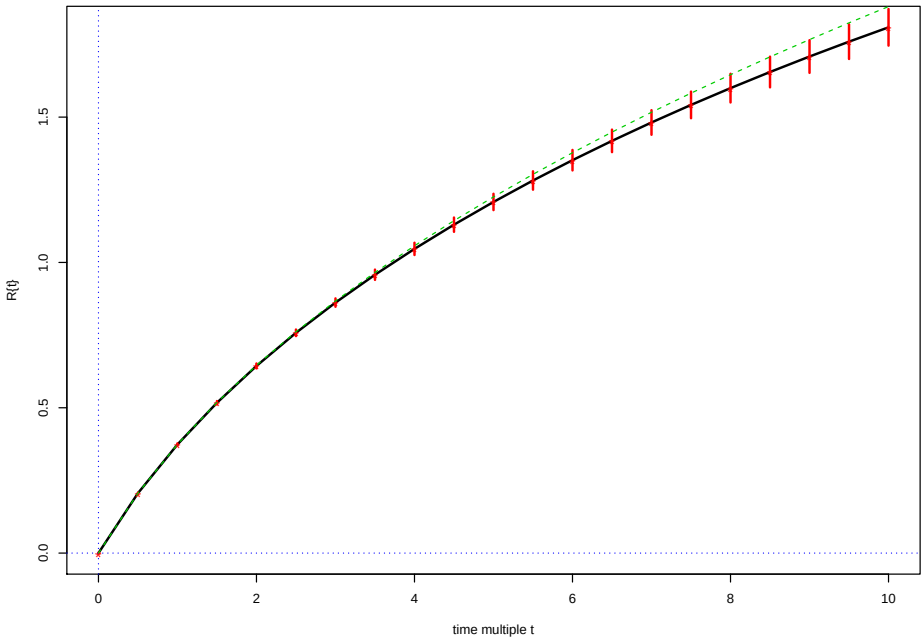


Figure 4: Predicted ratio of distinct new words found in t newly discovered Shakespeare canons, relative to the observed number $N = 30,688$ already seen. Bars indicate $\pm$ one standard error, as derived from (28) and (41). Light dashed line shows predictions from Fisher's gamma model (43)–(44).

4. A Guide to a new package deconvolveR

The 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 Q is normalized, `deconv` also accepts user-specified Q and 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 (20). 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, leading with the Shakespeare example.

4.1. 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.

```
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
[1,] 0.0183 0.00178 0.000151 0.00178 0.000151 0.000142
[2,] 0.0188 0.00178 0.000151 0.00356 0.000302 0.000142
[3,] 0.0193 0.00178 0.000150 0.00534 0.000452 0.000141
[4,] 0.0197 0.00179 0.000150 0.00713 0.000601 0.000141
[5,] 0.0202 0.00179 0.000149 0.00892 0.000751 0.000140
[6,] 0.0208 0.00179 0.000149 0.01071 0.000899 0.000140
R> tail(stats)
       theta        g     SE.g     G     SE.G   Bias.g
[336,]  79.4 0.000923 4.75e-05 0.995 2.87e-04 5.20e-06
[337,]  81.5 0.000916 5.06e-05 0.996 2.36e-04 4.85e-06
[338,]  83.5 0.000910 5.38e-05 0.997 1.82e-04 4.48e-06
[339,]  85.6 0.000903 5.73e-05 0.998 1.25e-04 4.11e-06
[340,]  87.8 0.000897 6.08e-05 0.999 6.45e-05 3.73e-06
[341,]  90.0 0.000891 6.45e-05 1.000      NaN 3.34e-06
```

By default, `deconv` assumes a Poisson family and works on a sample at a time. The above invocation provided $\mathcal{T}$, the sufficient statistic y (rather than the actual sample X) 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 (24).

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

```
R> print(result$S)
[1] 0.005534954
```

This indicates that the penalty term c_0 is not too big compared to the observed data.

4.2. A Poisson Simulation

θ	$g(\theta)$	Mean	StdDev	Bias	CoefVar
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 $Poisson(\Theta_i)$ for $i = 1, 2, \ldots, 1000$. The middle 4 columns have been multiplied by 100.

We next consider a simulation experiment. 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.

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

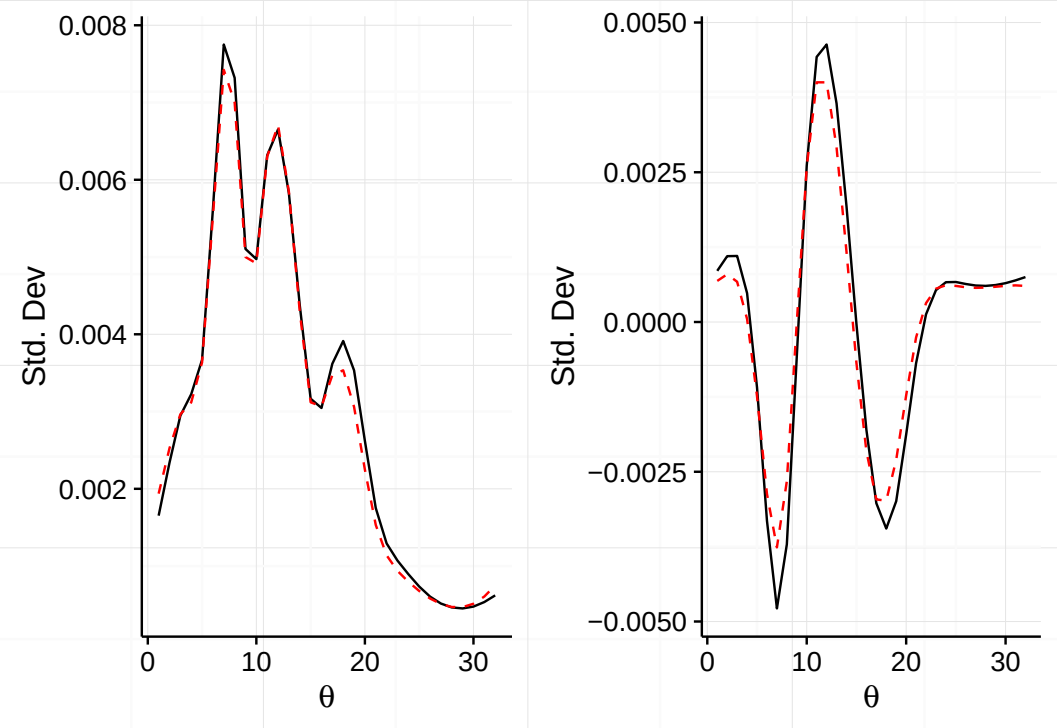


Figure 5: Standard deviations and biases for the simulated Poisson example. Solid curves are from the formulas and the dashed curves are from simulation.

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.

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

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

We've once again relied on the default *Poisson* family and a natural cubic spline basis of degree 5 for Q. The columns of 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.

```
stats <- sapply(results, function(x) x$stats$mat[, "g"])
mean <- apply(stats, 1, mean); sd <- apply(stats, 1, sd)
gTheta <- pchisq(tau, df = 10) - pchisq(c(0, tau[-length(tau)]), df = 10)
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 and biases of $g(\hat{\alpha})$ with the approximation given by the formulas in the paper for a few chosen values of Θ. The standard deviations are approximated well enough, but the biases are underestimated by the formulas.

4.3. A Normal Model

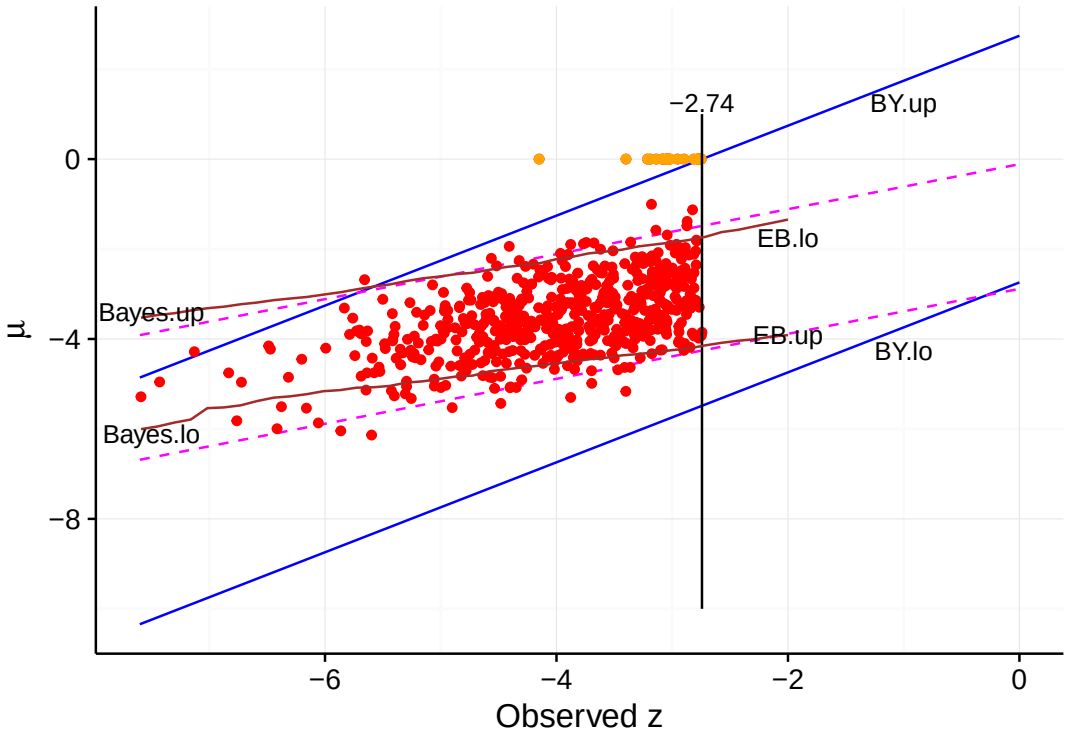


Figure 6: Estimates from g-modeling for a normal model along with 95% credible intervals (EB.up and EB.lo). They are a close match to the actual Bayes intervals.

Next, we consider data generated as follows:

$$z_i \sim N(\mu_i, 1) \text{ where } \mu_i = \begin{cases} 0, & \text{with probability } .9 \\ N(-3,1), & \text{with probability } .1 \end{cases} \text{ for } i = 1, 2, \ldots, 10,000. \quad (47)$$

To deconvolve this data, we specify an atom at zero using the parameter `deltaAt` which applies only to the normal case. Using $\tau = (-6, -5.75, \ldots, 3)$ 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.

```
tau <- seq(from = -6, to = 3, by = 0.25)
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.

Figures 7(a) and 7(b) investigate the effect of degrees of freedom and regularization in an example where the distribution of θ (here μ) is bimodal. Estimates $\hat{g}$ for degrees of freedom ranging from 2 to 6 superimposed on the histograms are shown in 7(a). A choice of 5 or 6 appears reasonable to capture the bimodality. Figure 7(b) shows the effect of the regularization parameter c_0 on the estimates: larger values for c_0 smooth out the $\hat{g}$ making the bimodality less prominent.

4.4. A Binomial 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 $X_i \sim Binomial(n_i, \theta_i)$ with θ_i being the probability of any one satellite site being malignant for the ith patient.

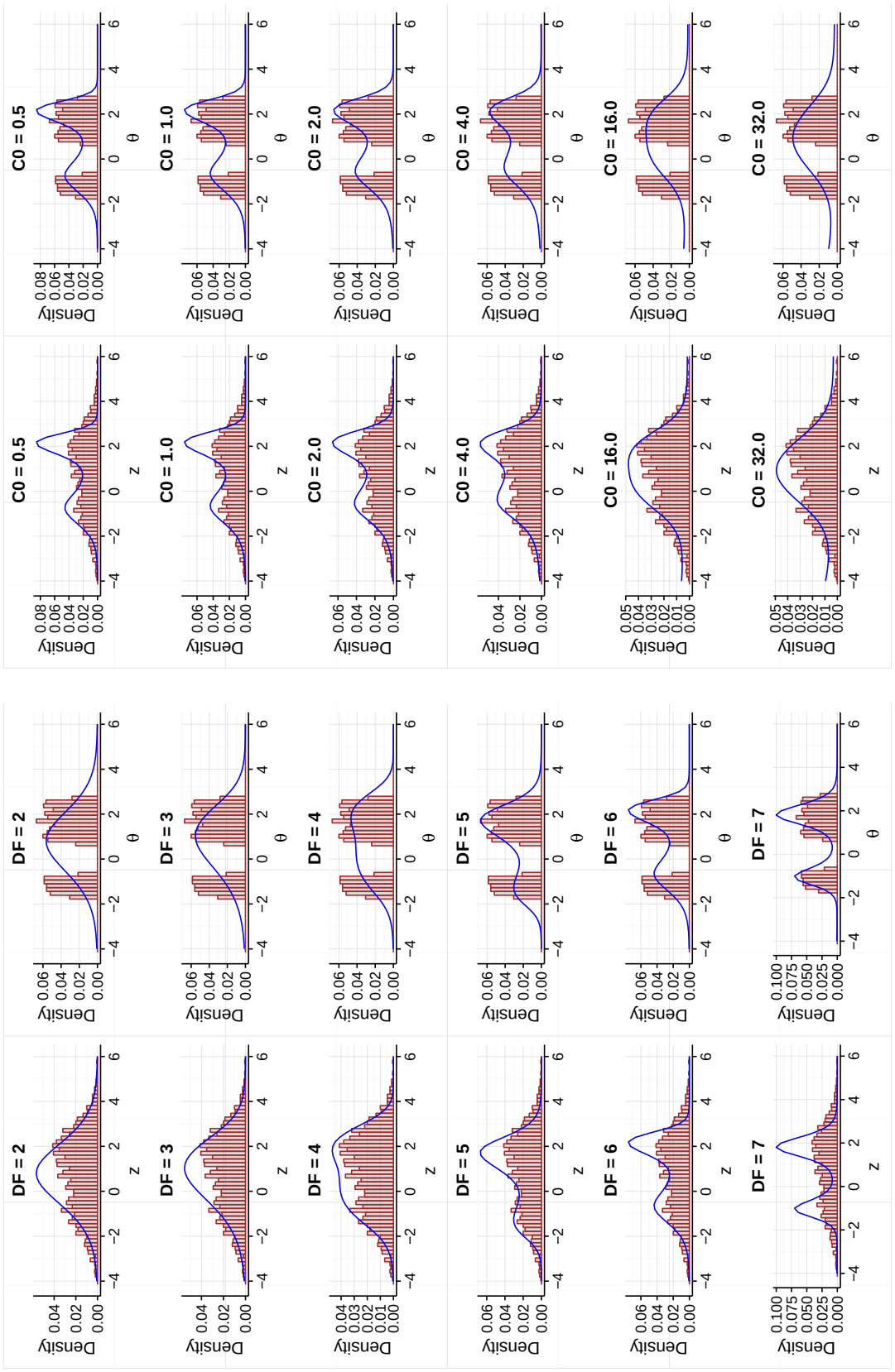
We take $\tau = (0.01, 0.02, \ldots, 0.09)$ (so $m = 99$), Q to be a 5-degree natural spline with columns standardized to mean 0 and sum of squares equal to 1 and the penalization parameter at the default value 1.

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

θ	$\hat{g}(\theta)$	SD (formula)	SD (simul.)	Bias (formula)	Bias (simul.)
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 3: Comparison of theoretical and bootstrap estimates of standard error for the surgery data using a binomial model. All values except the first column have been multiplied by 100.

Figure 8 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 at 50% chance of $\Theta <= 0.09$ and the remaining 50% spread out almost evenly over $[0.1, 1.0]$.



(a) The effect of the (spline basis) degrees of freedom on $\hat{g}$.

(b) The effect of varying c_0 penalty on $\hat{g}$.

Figure 7: The effect of the degrees of freedom and the penalty on the estimates

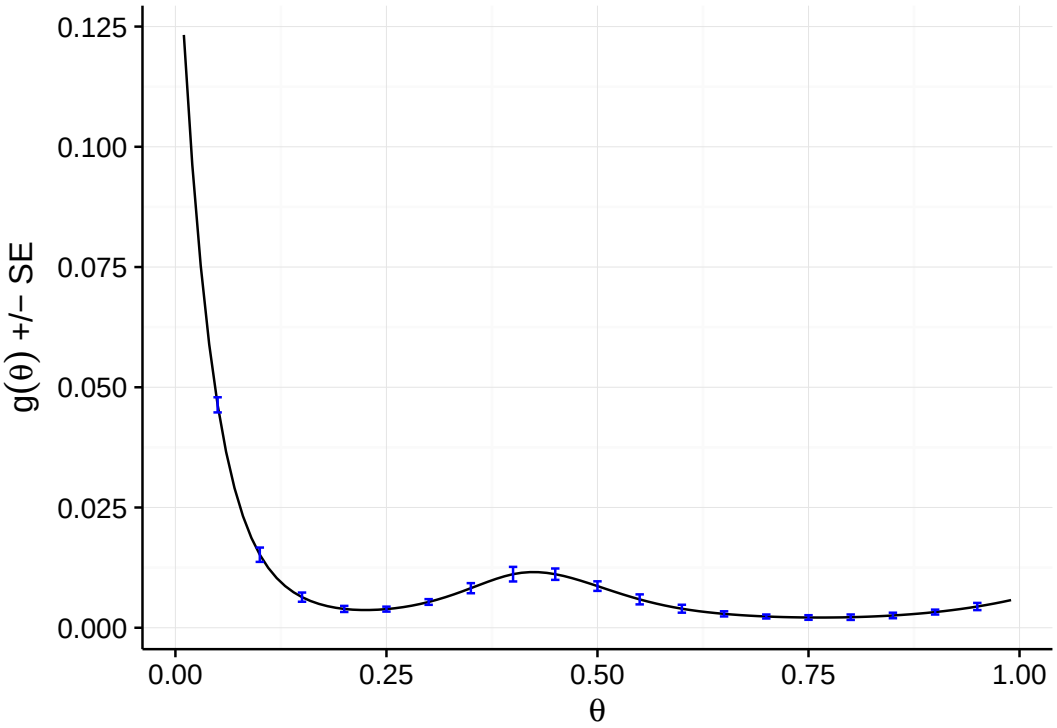


Figure 8: Estimates from g-modeling for a normal model along with 95% credible intervals. They are a close match to the actual Bayes intervals.

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 $\hat{\Theta}^*$ are sampled from the density $\hat{g}$. Each gave an $X_i \sim Binomial(n_i, \hat{\Theta}^*)$ with n_i the ith sample in the original data set. Finally, $\hat{\alpha}^*$ was computed using `deconv`. The results are shown in table Table 3.

5. Acknowledgements

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

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. ISBN 978-0-521-19249-1.

Efron B (2014). "Two Modeling Strategies for Empirical Bayes Estimation." *Statistical Science*, **29**(2), 285–301. ISSN 0883-4237. doi:10.1214/13-STS455.

Efron B (2016). "Empirical Bayes Deconvolution Estimates." *Biometrika*, **103**(1), 1–20. ISSN 0006-3444. doi:10.1093/biomet/asv068. URL http://biomet.oxfordjournals.org/content/103/1/1.full.pdf+html.

Efron B, Thisted R (1976). "Estimating the Number of Unseen Species: How Many Words Did Shakespeare Know?" *Biometrika*, **63**(3), 435–447. ISSN 0006-3444.

R Core Team (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Thisted R, Efron B (1987). "Did Shakespeare Write a Newly-Discovered Poem?" *Biometrika*, **74**(3), 445–455. ISSN 0006-3444.

Affiliation:

Balasubramanian Narasimhan
Department of Statistics
Sequoia Hall
Stanford University
390 Serra Mall
E-mail: naras@stanford.edu
URL: http://statistics.stanford.edu/~naras/

Bradley Efron
Department of Statistics
Sequoia Hall
Stanford University
390 Serra Mall
E-mail: brad@stat.stanford.edu
URL: http://efron.web.stanford.edu