Fully Bayes Normal Means

Matthew Stephens

May 3, 2018

Last updated: 2018-05-03

workflowr checks: (Click a bullet for more information)

- ▶ ✓ **R Markdown file:** up-to-date
- ▶ ✓ **Environment:** empty
- ▶ ✓ **Seed:** `set.seed(20180411)`
- ▶ ✓ **Session information:** recorded
- ▶ ✓ **Repository version:** 196d0e3 (https://github.com/stephens999/stat34800/tree/196d0e34e4c5f85715f675878331d6554b754cb0)

▶ **Expand here to see past versions:**

Background

In a previous homework you implemented Empirical Bayes (EB) shrinkage for the normal means problem with a normal prior. That is we have data $X = (X_1, \dots, X_n)$:

$$X_j|\theta_j, s_j \sim N(\theta_j, s_j^2)$$

and assume

$$\theta_j|\mu, \sigma \sim N(\mu, \sigma^2) \quad j = 1, \dots, n.$$

The EB approach involved two steps:

1. Estimates μ, σ by maximizing the log-likelihood $l(\mu, \sigma) = \log p(X|\mu, \sigma)$.
2. Compute the posteriore distribution $p(\theta_j|\hat{\mu}, \hat{\sigma})$.

The EB approach can be criticized for ignoring uncertainty in the estimates of μ and σ. Here we will use MCMC to do a fully Bayesian analysis that takes account of this uncertainty.

Fully Bayes approach

To make this easier we will first re-parameterize to use $\eta = log(\sigma)$, so η can take any value on the real line.

We will use a uniform prior on (μ, η), $p(\mu, \eta) \propto 1$ in the range $\mu = [-a, a]$ and $\eta \in [-b, b]$. You can use $a = 10^6$ and $b = 10$. (Because η is on the log scale, $b = 10$ covers a wide range of possible standard deviations). Thus the posterior distribution on μ, η is given by

$$p(\mu, \eta|X) \propto p(X|\mu, \eta)I(|\mu| < a)I(|\eta| < b)$$

where I denotes an indicator function.

1. Modify your log-likelihood computation code from your previous homework to compute the log-likelihood for (μ, η) given data X (and standard deviations s).
2. Use this to implement a MH algorithm to sample from $\pi(\mu, \eta) \propto p(X|mu, \eta)$. Note: In computing the MH acceptance probability you need to compute a ratio L_1/L_2. For numerical stability reasons you should always compute this ratio by $\exp(l_1 - l_2)$ where $l_i = \log(L_i)$ rather than directly computing L_1 and L_2 and then computing their ratio. (If both L_1 and L_2 are very small, they may be 0 to machine precision, which causes problems if you try to compute L_1/L_2 directly.)
3. Apply your MH algorithm to simulated data where you know the answer. Run you MH algorithm multiple (at least 3) times from multiple different initializations. For each run plot how the value of $log\pi(\mu^t, \eta^t)$ changes with iteration t. You should see that it starts from a low value (assuming you initialized to something that is not consistent with the data) and then gradually increases until it settles down to a "steady state" behavior. Use these plots to help decide how many iterations to run your algorithm to get reliable results (ie so results from different runs look similar) and how many iterations to discard as "burn-in". Compare your posterior distributions of μ and η with the true values you simulated (the distributions should cover the true values unless you did something wrong or are unlucky!) 1 2 3
4. Repeat part 3 for the "8 schools data" here (http://andrewgelman.com/2014/01/21/everything-need-know-bayesian-statistics-learned-eight-schools/) (omitting the comparisons with the true values, which of course you do not know here).
5. Note that the posterior distribution on θ_j is given by:

$$p(\theta_j|X) = \int p(\theta_j|X, \mu, \eta)p(\mu, \eta|X)$$

which is the expectation of $p(\theta_j|X, \mu, \eta)$ over the posteriore $p(\mu, \eta|X)$. Computing posterior distributions like this is sometimes referred to as "integrating out uncertainty in" μ, η. (It is useful to compare this with the EB approach of just plugging in the maximum likelihood estimates and computing $p(\theta_j|X, \hat{\mu}, \hat{\eta})$. Notice that the two will produce similar results if the posterior distribution $p(\mu, \eta|X)$ is very concentrated around the mle.)

Given T samples $\mu^1, \eta^1, \ldots, \mu^T, \eta^T$ from the posterior distribution $p(\mu, \eta|X)$ you can approximate this expectation by

$$p(\theta_j|X) \approx (1/T) \sum_t p(\theta_j|X, \mu^t, \eta^t).$$

So you can approximate the posterior mean by

$$E(\theta_j|X) \approx (1/T) \sum_t E(\theta_j|X, \mu^t, \eta^t).$$

Using the same idea, given an expression to approximate the posterior second moment $E(\theta_j^2|X)$, and so approximate the posterior variance (and hence posterior standard deviation).

6. Use the results from 4 and 5 to compute an approximate posterior mean and posterior standard deviation for θ_j for each school in the 8 schools data. Compare and contrast your results with the EB results and also the discussion in the initial blog-post here (http://andrewgelman.com/2014/01/21/everything-need-know-bayesian-statistics-learned-eight-schools/)

Session information

```
sessionInfo()
```

```
R version 3.3.2 (2016-10-31)
Platform: x86_64-apple-darwin13.4.0 (64-bit)
Running under: OS X El Capitan 10.11.6

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

loaded via a namespace (and not attached):
 [1] workflowr_1.0.1   Rcpp_0.12.16      digest_0.6.15
 [4] rprojroot_1.3-2   R.methodsS3_1.7.1 backports_1.1.2
 [7] git2r_0.21.0      magrittr_1.5      evaluate_0.10.1
[10] stringi_1.1.7     whisker_0.3-2     R.oo_1.22.0
[13] R.utils_2.6.0     rmarkdown_1.9     tools_3.3.2
[16] stringr_1.3.0     yaml_2.1.18       htmltools_0.3.6
[19] knitr_1.20
```

This reproducible R Markdown (http://rmarkdown.rstudio.com) analysis was created with workflowr (https://github.com/jdblischak/workflowr) 1.0.1
