13: Modal And Distributional Approximations

Taylor

University of Virginia

Introduction

We mention:

1. a few ways to find the posterior mode
2. how to approximate a posterior using a mode
3. slightly more involved ways to approximate your posterior

Newton's Method aka the Newton-Raphson algorithm

Based on a first-order approximation of the first derivative of the log-likelihood.

Approximate $L'(\theta) = (\log p(\theta \mid y))'$ as

$$\mathbf{0} \overset{\text{set}}{=} L'(\theta + \delta\theta) \approx L'(\theta) + L''(\theta)(\delta\theta)$$

rearranges to

$$\delta\theta = -[L''(\theta)]^{-1} L'(\theta)$$

Newton's Method

Repeat the following iteration until convergence:

$$\theta^t = \theta^{t-1} - [L''(\theta^{t-1})]^{-1} L'(\theta^{t-1})$$

Newton's Method aka the Newton-Raphson algorithm

Newton's Method

Repeat the following iteration until convergence:

$$\theta^t = \theta^{t-1} - [L''(\theta^{t-1})]^{-1} L'(\theta^{t-1})$$

Notes:

1. easily handles unnormalized densities
2. starting value is important because it is not guaranteed to converge from everywhere
3. The derivatives can be determined analytically or numerically

Quasi-Newton and conjugate gradient methods

Notes:

1. Quasi-Newton methods (approximate second derivatives) are available when second derivatives are too costly or unavailable
2. "Broyden-Fletcher-Goldfarb-Shanno" is a common example of a Quasi-Newton method
3. in R: `optim(2.9,F,method="BFGS")`
4. conjugate-gradient methods only use gradient information, but they are for models of the form $||A\theta - b||_2$ (also handled by `optim()`)
5. compared with the two above, they generally require more iterations, but use less computation per iteration and less storage

Numerical computation of derivatives

In `optim`, if you don't provide a function to calculate the gradient, then it uses a finite-difference approximation:

$$L_i'(\theta) = \frac{dL}{d\theta_i} \approx \frac{L(\theta + \delta_i e_i) - L(\theta - \delta_i e_i)}{2\delta_i}$$

and

$$\begin{aligned} L_{ij}''(\theta) &= \frac{d^2 L}{d\theta_i d\theta_j} \\ &\approx \frac{L_i'(\theta + \delta_j e_j) - L_i'(\theta - \delta_j e_j)}{2\delta_j} \end{aligned}$$

where e_j is the vector of all zeros except for a 1 in the jth spot, and δ_j is a small number (`optim`'s default is $1e-3$)

Gaussian approximations

Once the mode or modes have been found (perhaps after including a boundary-avoiding prior distribution as discussed in section 13.2, or after transforming the parameters appropriately), we can construct an approximation based on the multivariate normal distribution.

Let $\hat{\theta}$ be the mode, then

$$p(\theta \mid y) \approx N(\hat{\theta}, V_\theta)$$

where

$$V_\theta = \left[- \frac{d^2 \log p(\theta \mid y)}{d\theta^2} \bigg|_{\theta = \hat{\theta}} \right]^{-1}$$

is calculated exactly or approximated using the formula from a few slides ago.

Example

From chapter 3:

1. $p(y_i \mid \mu, \sigma^2) = \text{Normal}(\mu, \sigma^2)$
2. $p(\mu, \sigma^2) \propto 1/\sigma^2$

$$p(\mu, \sigma^2 \mid y) \propto (\sigma^2)^{-(n+2)/2} \exp\left[-\frac{1}{2\sigma^2}\left\{(n-1)s^2 + n(\bar{y}-\mu)^2\right\}\right]$$

Example

$$p(\mu, \sigma^2 \mid y) \propto (\sigma^2)^{-(n+2)/2} \exp\left[-\frac{1}{2\sigma^2}\left\{(n-1)s^2 + n(\bar{y}-\mu)^2\right\}\right]$$

Letting $\theta = \log\sigma$, $p(\mu, \theta \mid y)$ is proportional to

$$\exp[-n\theta]\exp\left[-\frac{1}{2\exp[2\theta]}\left\{(n-1)s^2 + n(\bar{y}-\mu)^2\right\}\right]$$

So $\log p(\mu, \theta \mid y)$ is

$$constant - n\theta - .5\exp(-2\theta)\left[(n-1)s^2 + n(\bar{y}-\mu)^2\right]$$

and $L'(\theta) = \begin{bmatrix} \exp(-2\theta)(\bar{y}-\mu)n \\ -n + \exp(-2\theta)\left[(n-1)s^2 + n(\bar{y}-\mu)^2\right] \end{bmatrix}$

Example

Warning: `optim` *minimizes*, so we use $-\log p(\mu, \theta \mid y)$

$$n\theta + .5\exp(-2\theta)\left[(n-1)s^2 + n(\bar{y}-\mu)^2\right]$$

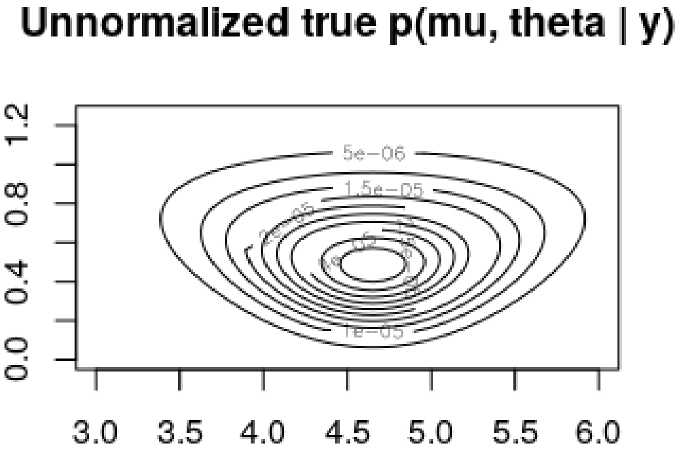
and

$$L'(\theta) = \begin{bmatrix} -\exp(-2\theta)(\bar{y}-\mu)n \\ n - \exp(-2\theta)\left[(n-1)s^2 + n(\bar{y}-\mu)^2\right] \end{bmatrix}$$

```
# if gr left blank, finite difference approx. used
optim_results <- optim(par = c(5, 0),
                       fn = neg_log_unnorm_post,
                       gr = gradient,
                       method = "BFGS",
                       hessian = T)
```

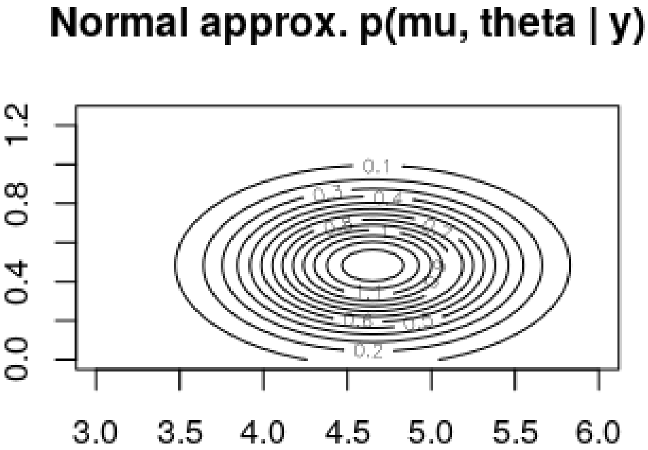
See `mode_finding_examples.r`

Example



Unnormalized true p(mu, theta | y)
5e−06
1.5e−05
1e−05
0.0
0.4
0.8
1.2
3.0
3.5
4.0
4.5
5.0
5.5
6.0

Example



Gaussian approximations: Laplace's Method

If you want approximations to posterior *expectations* (say $E[h(\theta) \mid y]$), then you might consider Laplace's method, which is based on second-order Taylor approximations of the functions:

1. $u_1(\theta) = \log[h(\theta)q(\theta \mid y)]$
2. $u_2(\theta) = \log q(\theta \mid y)$

where $p(\theta \mid y) = q(\theta \mid y) / \int q(\theta \mid y) \mathrm{d}\theta$.

Both are centered at maximizing values: θ_0^1, θ_0^2, and this assumes hs are twice continuously differentiable.

Idea:

$$\frac{\int h(\theta)q(\theta \mid y)\mathrm{d}\theta}{\int q(\theta \mid y)\mathrm{d}\theta} = \frac{\int \exp\left[\log h(\theta) + \log q(\theta \mid y)\right] \mathrm{d}\theta}{\int \exp\left[\log q(\theta \mid y)\right] \mathrm{d}\theta}$$

Gaussian approximations: Laplace's Method

Exponentiating and integrating (typo on page 318?)

$$
\begin{aligned}
u(\theta) &\approx u(\theta_0) + (\theta - \theta_0)^T u'(\theta_0) + \frac{1}{2}(\theta - \theta_0)^T u''(\theta_0)(\theta - \theta_0) \\
&= u(\theta_0) + \frac{1}{2}(\theta - \theta_0)^T u''(\theta_0)(\theta - \theta_0)
\end{aligned}
$$

gives us

$$
\begin{aligned}
\int \exp[u(\theta)] \mathrm{d}\theta &\approx \int \exp[u(\theta_0) + \frac{1}{2}(\theta - \theta_0)^T u''(\theta_0)(\theta - \theta_0)] \mathrm{d}\theta \\
&= \exp[u(\theta_0)] \int \exp\left[\frac{1}{2}(\theta - \theta_0)^T u''(\theta_0)(\theta - \theta_0)\right] \mathrm{d}\theta \\
&= \exp[u(\theta_0)] \int \exp\left[-\frac{1}{2}(\theta - \theta_0)^T \{-u''(\theta_0)\}(\theta - \theta_0)\right] \mathrm{d}\theta \\
&= \exp[u(\theta_0)](2\pi)^{d/2} \det[-u''(\theta_0)]^{-1/2}
\end{aligned}
$$

Gaussian approximations

The book has a few more generalizations that we don't address:

1. approximating multimodal distributions with normal mixtures
2. approximating multimodal distributions with t mixtures

The EM Algorithm

The **expectation-maximization algorithm** finds the argument that maximizes the marginal posterior. It's useful in situations where there is missing data in a model (e.g. factor models, hidden markov models, state space models, etc.).

It folows the following steps

1. replace missing values by their expectations given the guessed parameters,
2. estimate parameters assuming the missing data are equal to their estimated values,
3. re-estimate the missing values assuming the new parameter estimates are correct,
4. re-estimate parameters,

and so forth, iterating until convergence.

The EM Algorithm

Call $\theta = (\gamma, \phi)$. You're interested in the mode of $p(\phi \mid y)$. Typically, γ is "hidden data."

$$\log p(\phi \mid y) = \log \frac{p(\gamma, \phi \mid y)}{p(\gamma \mid \phi, y)} = \log \underbrace{p(\gamma, \phi \mid y)}_{\text{joint posterior}} - \log \underbrace{p(\gamma \mid \phi, y)}_{\text{conditional posterior}}$$

The EM Algorithm

Call $\theta = (\gamma, \phi)$. You're interested in the mode of $p(\phi \mid y)$. Typically, γ is "hidden data."

$$\log p(\phi \mid y) = \log \frac{p(\gamma, \phi \mid y)}{p(\gamma \mid \phi, y)} = \log \underbrace{p(\gamma, \phi \mid y)}_{\text{joint posterior}} - \log \underbrace{p(\gamma \mid \phi, y)}_{\text{conditional posterior}}$$

taking expectations on both sides with respect to $p(\gamma \mid \phi^{\text{old}}, y)$ yields:

$$\log p(\phi \mid y) = E\left[\log p(\gamma, \phi \mid y) \mid \phi^{\text{old}}, y\right] - E\left[\log p(\gamma \mid \phi, y) \mid \phi^{\text{old}}, y\right]$$

The EM Algorithm

We iteratively use the middle term in
$\log p(\phi \mid y) = E\left[\log p(\gamma, \phi \mid y) \mid \phi^{\text{old}}, y\right] - E\left[\log p(\gamma \mid \phi, y) \mid \phi^{\text{old}}, y\right]$.

The Q quantity in the "E" step

$$Q(\phi \mid \phi^{\text{old}}) = E\left[\log p(\gamma, \phi \mid y) \mid \phi^{\text{old}}, y\right]$$

The EM Algorithm

We iteratively use the middle term in
$\log p(\phi \mid y) = E\left[\log p(\gamma, \phi \mid y) \mid \phi^{\text{old}}, y\right] - E\left[\log p(\gamma \mid \phi, y) \mid \phi^{\text{old}}, y\right].$

The Q quantity in the "E" step

$$Q(\phi \mid \phi^{\text{old}}) = E\left[\log p(\gamma, \phi \mid y) \mid \phi^{\text{old}}, y\right]$$

The EM algorithm

Repeat the following until convergence:

1. E-step: calculate $Q(\phi \mid \phi^{\text{old}})$
2. M-step: replace ϕ^{old} with $\arg\max Q(\phi \mid \phi^{\text{old}})$

The EM Algorithm

If we follow this strategy, $\log p(\phi \mid y)$ increases at every iteration:

$$
\begin{aligned}
\log p(\phi \mid y) &= E\left[\log p(\gamma, \phi \mid y) \mid \phi^{\text{old}}, y\right] - E\left[\log p(\gamma \mid \phi, y) \mid \phi^{\text{old}}, y\right] \\
&= Q(\phi \mid \phi^{\text{old}}) - E\left[\log p(\gamma \mid \phi, y) \mid \phi^{\text{old}}, y\right] && \text{(defn. Q)} \\
&\geq Q(\phi \mid \phi^{\text{old}}) - E\left[\log p(\gamma \mid \phi^{\text{old}}, y) \mid \phi^{\text{old}}, y\right] && \text{(HW)}
\end{aligned}
$$

The EM Algorithm

If we follow this strategy, $\log p(\phi \mid y)$ increases at every iteration:

$$\begin{aligned}
\log p(\phi \mid y) &= E\left[\log p(\gamma, \phi \mid y) \mid \phi^{\text{old}}, y\right] - E\left[\log p(\gamma \mid \phi, y) \mid \phi^{\text{old}}, y\right] \\
&= Q(\phi \mid \phi^{\text{old}}) - E\left[\log p(\gamma \mid \phi, y) \mid \phi^{\text{old}}, y\right] && \text{(defn. Q)} \\
&\geq Q(\phi \mid \phi^{\text{old}}) - E\left[\log p(\gamma \mid \phi^{\text{old}}, y) \mid \phi^{\text{old}}, y\right] && \text{(HW)}
\end{aligned}$$

So

$$\begin{aligned}
&\log p(\phi^{\text{new}} \mid y) - \log p(\phi^{\text{old}} \mid y) \\
&= \log p(\phi^{\text{new}} \mid y) - \left\{ Q(\phi^{\text{old}} \mid \phi^{\text{old}}) - E\left[\log p(\gamma \mid \phi^{\text{old}}, y) \mid \phi^{\text{old}}, y\right] \right\} \\
&\geq Q(\phi^{\text{new}} \mid \phi^{\text{old}}) - E\left[\log p(\gamma \mid \phi^{\text{old}}, y) \mid \phi^{\text{old}}, y\right] \\
&\qquad - \left\{ Q(\phi^{\text{old}} \mid \phi^{\text{old}}) - E\left[\log p(\gamma \mid \phi^{\text{old}}, y) \mid \phi^{\text{old}}, y\right] \right\} \\
&= Q(\phi^{\text{new}} \mid \phi^{\text{old}}) - Q(\phi^{\text{old}} \mid \phi^{\text{old}})
\end{aligned}$$

The EM Algorithm

Notes:

1. The EM algo isn't inherently Bayesian. It can also be used to accomplish maximum likelihood estimation.
2. The expectation of $\log p(\gamma, \phi \mid y)$ is usually easy to compute because it is a sum, and might only depend on sufficient statistics
3. The EM algorithm implicitly deals with parameter constraints in the M-step
4. The EM algorithm is parameterization independent
5. The *Generalized* EM (GEM) just increases Q instead of maximizing it.
6. The book describes many generalizations, in addition to this one
7. You might find multiple modes if you start from multiple starting points (using mixture approximations afterwards)
8. if you can, debug by printing $\log p(\phi^i \mid y)$ at every iteration and make sure it increases monotonically

Variational inference approximates an intractable posterior $p(\theta \mid y)$ with some chosen distribution $g(\theta \mid \phi)$ (e.g. multivariate normal).

Variational Inference

Variational inference approximates an intractable posterior $p(\theta \mid y)$ with some chosen distribution $g(\theta \mid \phi)$ (e.g. multivariate normal).

We will assume this approximating distribution factors into J components:

$$g(\theta \mid \phi) = \prod_{j=1}^{J} g_j(\theta_j \mid \phi_j) = g_j(\theta_j \mid \phi_j) g_{-j}(\theta_{-j} \mid \phi_{-j}).$$

We will find ϕ using an EM-like algorithm that minimizes Kullback-Leibler divergence.

Variational Inference

Kullback-Leibler divergence is "reversed" this time:

$$
\begin{aligned}
KL(g||p) &= -\int \log\left(\frac{p(\theta \mid y)}{g(\theta \mid \phi)}\right) g(\theta \mid \phi)\mathrm{d}\theta \\
&= -\int \log\left(\frac{p(\theta, y)}{g(\theta \mid \phi)}\right) g(\theta \mid \phi)\mathrm{d}\theta + \int \log p(y) g(\theta \mid \phi)\mathrm{d}\theta \\
&= \underbrace{-\int \log\left(\frac{p(\theta, y)}{g(\theta \mid \phi)}\right) g(\theta \mid \phi)\mathrm{d}\theta}_{\text{variational lower bound}} + \log p(y)
\end{aligned}
$$

The term that we maximize (minimize the negative) is called the **variational lower bound** aka the **evidence lower bound** (ELBO).

Variational Inference

Every iteration, we cycle through all the hyper-parameters $\phi_1, \ldots, \phi_J$, and change them until convergence is reached.

Variational Inference

Every iteration, we cycle through all the hyper-parameters $\phi_1, \ldots, \phi_J$, and change them until convergence is reached.

Looking at ϕ_j...

$$\int \log\left(\frac{p(\theta, y)}{g(\theta \mid \phi)}\right) g(\theta \mid \phi) \mathrm{d}\theta$$

$$= \iint \left[\log p(\theta, y) - \log g_j(\theta_j \mid \phi_j) - \log g_{-j}(\theta_{-j} \mid \phi_{-j})\right]$$
$$g_j(\theta_j \mid \phi_j) g_{-j}(\theta_{-j} \mid \phi_{-j}) \mathrm{d}\theta_j \mathrm{d}\theta_{-j}$$

$$= \int \left[\int \log p(\theta, y) g_{-j}(\theta_{-j} \mid \phi_{-j}) \mathrm{d}\theta_{-j}\right] g_j(\theta_j \mid \phi_j) \mathrm{d}\theta_j$$

$$- \int \log g_j(\theta_j \mid \phi_j) g_j(\theta_j \mid \phi_j) \mathrm{d}\theta_j - \int \log g_{-j}(\theta_{-j} \mid \phi_{-j}) g_{-j}(\theta_{-j} \mid \phi_{-j}) \mathrm{d}\theta_{-j}$$

$$= \int \log\left(\frac{\tilde{p}(\theta_j)}{g_j(\theta_j \mid \phi_j)}\right) g_j(\theta_j \mid \phi_j) \mathrm{d}\theta_j + \text{constant} \qquad (*)$$

Variational Inference

We think of $\tilde{p}(\theta_j)$ as an unnormalized density

$$\log \tilde{p}(\theta_j) = \int \log p(\theta, y) g_{-j}(\theta_{-j} \mid \phi_{-j}) \mathrm{d}\theta_{-j}$$

because usually

$$\begin{aligned}
\int \tilde{p}(\theta_j)\mathrm{d}\theta_j &= \int \exp\left[\int \log p(\theta, y) g_{-j}(\theta_{-j} \mid \phi_{-j}) \mathrm{d}\theta_{-j}\right] \mathrm{d}\theta_j \\
&\leq \int \exp\left[\log \int p(\theta, y) g_{-j}(\theta_{-j} \mid \phi_{-j}) \mathrm{d}\theta_{-j}\right] \mathrm{d}\theta_j \qquad \text{(Jensen's)} \\
&= \iint p(\theta, y) g_{-j}(\theta_{-j} \mid \phi_{-j}) \mathrm{d}\theta_{-j} \mathrm{d}\theta_j \\
&< \infty
\end{aligned}$$

Variational Inference

VI algorithm

For $j = 1, \ldots, J$ set ϕ_j so that $\log g_j(\theta_j \mid \phi_j)$ is equal to

$$\log \tilde{p}(\theta_j) = \int \log p(\theta, y) g_{-j}(\theta_{-j} \mid \phi_{-j}) \mathrm{d}\theta_{-j}$$

Variational Inference: educational testing example

When the parameters are $\alpha_1, \ldots, \alpha_8, \mu, \tau$, the log posterior is

$$\log p(\theta \mid y) = \text{constant} - \frac{1}{2}\sum_{j=1}^{8} \frac{(y_j - \alpha_j)^2}{\sigma_j^2} - 8 \log \tau - \frac{1}{2}\frac{1}{\tau^2}\sum_{j=1}^{8}(\alpha_j - \mu)^2$$

and we assume

$$g(\alpha_1, \ldots, \alpha_8, \mu, \tau) = g(\alpha_1) \times \cdots \times g(\alpha_8) g(\mu) g(\tau).$$

Let's reparameterize τ as τ^2 and assume $g(\alpha_1), \ldots, g(\alpha_8) g(\mu)$ are all normal distributions, and $g(\tau^2)$ is an Inverse-Gamma.

Variational Inference: example

$$
\begin{aligned}
&\log g(\alpha_j) \\
&\overset{\text{set}}{=} \log \tilde{p}(\alpha_j) \\
&= \int \log p(\theta, y) g_{-j}(\theta_{-j}) \mathrm{d}\theta_{-j} \\
&= -\frac{1}{2} \sum_{i=1}^{8} \frac{E_{-j}[(y_i - \alpha_i)^2]}{\sigma_i^2} - 8 E_{-j}[\log \tau] - \frac{1}{2} E_{-j}\left[\frac{1}{\tau^2}\right] \sum_{i=1}^{8} E[(\alpha_i - \mu)^2] + c \\
&= -\frac{1}{2} \frac{(y_j - \alpha_j)^2}{\sigma_j^2} - \frac{1}{2} E_{-j}\left[\frac{1}{\tau^2}\right] E_{-j}[(\alpha_j - \mu)^2] + c' \\
&= -\frac{1}{2} \frac{(y_j - \alpha_j)^2}{\sigma_j^2} - \frac{1}{2} E_{-j}\left[\frac{1}{\tau^2}\right] (\alpha_j^2 - 2\alpha_j E_{-j}[\mu])] + c''
\end{aligned}
$$

We are using linearity, independence, the data aren't random, and we're grouping all the terms that don't involve α_j into the constant.

Variational Inference: example

For μ:

$$\begin{aligned}
\log \tilde{p}(\mu) &= \int \log p(\theta, y) g_{-j}(\theta_{-j} \mid \phi_{-j}) \mathrm{d}\theta_{-j} \\
&= -\frac{1}{2} E_{-\mu} \left[\frac{1}{\tau^2} \sum_{j=1}^{8} (\alpha_j - \mu)^2 \right] + \text{constant} \\
&= -\frac{1}{2} E_{-\mu} \left[\frac{1}{\tau^2} \right] \sum_{j=1}^{8} \left(\mu^2 - 2\mu E_{-\mu}[\alpha_j] \right) + \text{constant} \\
&= -\frac{1}{2} E_{-\mu} \left[\frac{1}{\tau^2} \right] \left(8\mu^2 - 2\mu \sum_{j=1}^{8} E_{-\mu}[\alpha_j] \right) + \text{constant}
\end{aligned}$$

So $g(\mu) = \ldots$

Variational Inference: example

For τ (not τ^2):

$$\log \tilde{p}(\tau) = \int \log p(\theta, y) g_{-j}(\theta_{-j} \mid \phi_{-j}) \mathrm{d}\theta_{-j}$$

$$= -8 \log \tau - \frac{1}{2}\frac{1}{\tau^2} E_{-\tau}\left[\sum_{j=1}^{8} (\alpha_j - \mu)^2\right] + c$$

So $g(\tau) \propto \tau^{-8} \exp\left[-\frac{\sum_j E_{-\tau}[(\alpha_j - \mu)^2]}{2\tau^2}\right]$ which means

$$g(\tau^2) = (\tau^2)^{-(\frac{7}{2}+1)} \exp\left[-\frac{\sum_j E_{-\tau}[(\alpha_j - \mu)^2]}{2\tau^2}\right]$$

which is an InverseGamma$\left(\frac{7}{2}, \frac{\sum_j E_{-\tau}[(\alpha_j - \mu)^2]}{2}\right)$

Variational Inference: example

To complete this example, we need to derive:

- for $g(\alpha_j)$:
  1. $E_{-j}\left[\frac{1}{\tau^2}\right] = E_{\tau^2}\left[\frac{1}{\tau^2}\right]$,
  2. $E_{-j}[\mu] = E_{\mu}[\mu]$
- for $g(\mu)$:
  1. $E_{-\mu}[\alpha_j] = E_{\alpha_j}[\alpha_j]$,
  2. $E_{-j}\left[\frac{1}{\tau^2}\right] = E_{\tau^2}\left[\frac{1}{\tau^2}\right]$
- for $g(\tau^2)$:
  1. $\sum_j E_{-\tau}[(\alpha_j - \mu)^2] = \sum_j E_{\alpha_j,\mu}[(\alpha_j - \mu)^2]$,

Variational Inference: example

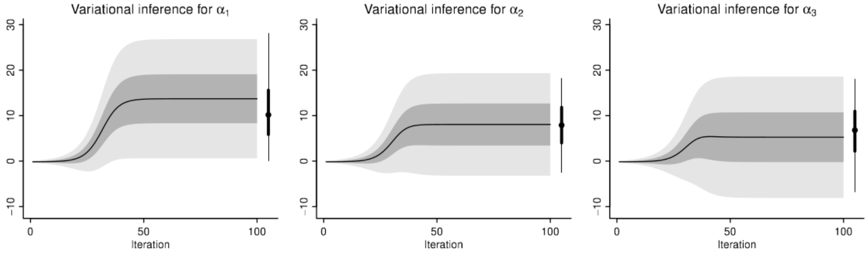


Figure 13.6 *Progress of inferences for the effects in schools A, B, and C, for 100 iterations of variational Bayes. The lines and shaded regions show the median, 50% interval, and 90% interval for the variational distribution. Shown to the right of each graph are the corresponding quantiles for the full Bayes inference as computed via simulation.*