Examples with VB, EM and MCMC algorithms

Variational Bayes is a deterministic algorithm that can address the computational limitations of sampling-based algorithms (e.g. Gibbs sampling or Metropolis Hastings) by using **approximate inference** (ATTIAS 2000; JAAKKOLA 2001). The approach was initially proposed in the machine learning literature (JORDAN *et al.* 1999); since then, the method has been applied for image data analysis and graphical modeling (HERMOSILLO *et al.* 2002; BLEI and JORDAN 2006). Recently, VB has been used in many applications in genetics studies, for instance to perform GWAS (LOGSDON *et al.* 2010; CARBONETTO and STEPHENS 2012; LOH *et al.* 2015), in the estimation of QTL effects with epistasis (LI and SILLANPÄÄ 2012), for heritability estimation with the G-BLUP (ARAKAWA 2014), for multi-trait analysis and prediction (HAYASHI and IWATA 2013), and for efficient inference of population structure (RAJ *et al.* 2014). The VB framework has been used to implement methods that perform shrinkage (LI and SILLANPÄÄ 2012; ARAKAWA 2014) and a combination of variable selection and shrinkage (CARBONETTO and STEPHENS 2012; HAYASHI and IWATA 2013) at an individual marker level.

MCMC gives *approximate* solutions to the *exact* posterior. If the algorithm converges the Monte Carlo error diminishes as the number of MCMC samples increases; however, in practice the number of MCMC samples used for inferences is finite and MCMC estimates are only approximations to the true posterior estimates. In contrast, VB can arrive at *exact* solutions to an *approximate* posterior. Deriving VB estimates usually involves an optimization and update procedure similar to the ones involved in the Expectation-Maximization (EM) algorithm (GELMAN *et al.* 2004; BISHOP 2006). Iterations in EM ultimately result in convergence to (local) MAP estimates (posterior modes) of the parameters. In contrast, iterations in VB result in an approximation of a closed-form density function that is *close* to the posterior, rather than of the posterior itself.

In a Bayesian model, the joint posterior density is given as the product of the likelihood function and the prior, i.e. $p(\boldsymbol{\beta}|\mathbf{y}) \propto p(\boldsymbol{y}|\boldsymbol{\beta})p(\boldsymbol{\beta})$, for a p-dimensional parameter vector $\boldsymbol{\beta}$ and data-vector $\mathbf{y}$. VB is applied when it is difficult to draw samples from the posterior $p(\boldsymbol{\beta}|\mathbf{y})$ and it is easier to handle some class of approximating distributions $q(\boldsymbol{\beta}|\boldsymbol{\phi})$ which is used to approximate the true posterior. Here, $\boldsymbol{\phi}$ represent the parameters of the variational approximation. In principle, VB finds $q(\boldsymbol{\beta}|\boldsymbol{\phi})$ such that the Kullback-Leibler (KL) distance (COVER and THOMAS 2012) between the approximating density and the true posterior density, $p(\boldsymbol{\beta}|\mathbf{y})$, is minimized. A standard approach in VB, referred as to mean-field approximation, is to structure $q(\boldsymbol{\beta})$ such that $q(\boldsymbol{\beta}) = \prod_{j=1}^{p} q_j(\beta_j|\phi_j)$ (BISHOP 2006). If the parameter vector $\boldsymbol{\beta}$ comprises p regression coefficients (e.g. marker effects), this condition assumes that the regression coefficients are independent of each other at $q(\boldsymbol{\beta})$. This factorized form lends VB its computational speed since time complexity now varies linearly with the number of markers (LOGSDON *et al.* 2010; CARBONETTO and STEPHENS 2012).

Let's consider two simple examples that demonstrates how the VB algorithm is applied.

I) UNIVARIATE GAUSSIAN

This example is taken from BISHOP 2006 pp. 470-476. Let's consider a Gaussian distribution with unknown mean and *precision*, denoted by μ and τ respectively. We infer the posterior distribution of these parameters given the data $\boldsymbol{y} = \{y_1, y_2, \dots y_N\}$. The likelihood function is then given by:

$$p(\boldsymbol{y}|\mu,\tau) = \left(\frac{\tau}{2\pi}\right)^{\frac{N}{2}} \exp\left\{-\left(\frac{\tau}{2}\right)\sum_{i=1}^{N}(y_i-\mu)^2\right\} \quad [1]$$

Correspondingly, the prior distributions are given by:

$$p(\mu|\tau) = \frac{\lambda_0\tau}{2\pi}\exp\left\{-\left(\frac{\lambda_0\tau}{2}\right)(\mu-\mu_0)^2\right\} = N(\mu|\mu_0,(\lambda_0\tau)^{-1}) \quad [2]$$

$$p(\tau) = \frac{1}{b_0^{a_0}\Gamma(a_0)}\exp\left\{-\left(\frac{\tau}{b_0}\right)\tau^{a_0-1}\right\} = Gamma(\tau|a_0,b_0) \quad [3]$$

Thus, the posterior distribution can be expressed as:

$$p(\mu,\tau|\boldsymbol{y}) \propto p(\boldsymbol{y}|\mu,\tau)p(\mu|\tau)p(\tau) \quad [4]$$

We can approximate the true posterior using its variational approximation denoted by $q(\mu,\tau)$, which can be expressed in a factorized form as follows:

$$q(\mu,\tau) = q_\mu(\mu)q_\tau(\tau) \quad [5]$$

It is to be noted that the true posterior does not approximate this way. Each of these two components is obtained by averaging over the parameter in the other component. We assume here that the parameters μ_0, τ_0, a_0, b_0 are known.

$$\log\left(q_\mu(\mu)\right) = E_\tau\left[\log\left(p(\boldsymbol{y}|\mu,\tau)\right)\right] + E_\tau[\log(p(\mu|\tau))] \quad [6]$$
$$= -\left[\frac{E_\tau(\tau)}{2}\right]\left\{\lambda_0(\mu-\mu_0)^2 + \sum_{i=1}^{N}(y_i-\mu)^2\right\} + c$$

Completing the square and solving, we get a Gaussian distribution with mean and precision given by μ_N and λ_N respectively, where,

$$\mu_N = \frac{\lambda_0\mu_0 + N\bar{y}}{\lambda_0 + N}$$
$$\lambda_N = (\lambda_0 + N)E(\tau)$$

Likewise,

$$\log\left(q_\tau(\tau)\right) = E_\mu\left[\log\left(p(\boldsymbol{y}|\mu,\tau)\right)\right] + E_\mu[\log(p(\mu|\tau))] + E_\mu[\log(p(\tau))] \quad [7]$$
$$= (a_0-1)\log\tau - b_0\tau + \left(\frac{1}{2}\log(\tau)\right) + \left(\frac{N}{2}\log(\tau)\right) - \frac{\tau}{2}E_\mu\left\{\lambda_0(\mu-\mu_0)^2 + \sum_{i=1}^{N}(y_i-\mu)^2\right\} + c$$

Upon solving, we see that $q_\tau(\tau)$ follows a Gamma distribution with parameters a_N and b_N respectively, where

$$a_N = a_0 + \frac{N+1}{2}$$

$$b_N = b_0 + \frac{1}{2}E_\mu\left[\lambda_0(\mu-\mu_0)^2 + \sum_{i=1}^{N}(y_i-\mu)^2\right] \quad [8]$$

$$= b_0 + \frac{1}{2}E_\mu(\mu^2)(N+\lambda_0) - 2E_\mu(\mu)\left(\sum_{i=1}^{N} y_i + \lambda_0\mu_0\right) + \sum_{i=1}^{N} y_i^2 + \lambda_0\mu_0^2$$

The optimal distributions $q_\mu(\mu)$ and $q_\tau(\tau)$ arose from the structure of the likelihood function and our choice of conjugate priors. We solve for this in an iterative manner by assuming an initial value for $E_\tau(\tau)$ and using this to re-compute the distribution $q_\mu(\mu)$. This will help us re-estimate the moments $E(\mu)$ and $E(\mu^2)$, which can be used to re-compute the distribution $q_\tau(\tau)$. We proceed iteratively in this manner until convergence occurs.

```
### VB for Univariate Gaussian

N = 1000
set.seed(100)
x <- rnorm(N)

mu00 = rnorm(1)
b00 = a00 = runif(1)
lambda00  = runif(1)

mu0 = rnorm(1)
b0 = runif(1)
a0 = runif(1)
lambda0  = runif(1)

reps=1000

lambda_N =  b_N = E_tau = E_mu = E_mu2 = q_mu = q_tau = KL = matrix(,reps,1)
E_tau[1,] = a0/b0

mu_N = ((lambda0*mu0)+(N*mean(x)))/(lambda0+N)
a_N = a0+((N+1)/2)

lambda_N[1,] = (lambda0+N)*E_tau[1,]
q_mu[1,] = rnorm(1,mu_N,lambda_N[1,])
E_mu[1,] = mu_N
E_mu2[1,] = (1/lambda_N[1,])+mu_N^2
b_N[1,] = b0+0.5*((E_mu2[1,])*(N+lambda0)-
(2*E_mu[1,]*((N*mean(x))+(lambda0*mu0)))+((lambda0*(mu0^2))+sum(x^2)))

q_tau[1,] = rgamma(1,a_N,b_N[1,])
```

```
for(i in 2:reps){

  lambda_N[i,] = (lambda0+N)*E_tau[i-1,]
  q_mu[i,] = rnorm(1,mu_N,lambda_N[i,])
  E_mu[i,] = mu_N
  E_mu2[i,] = (1/lambda_N[i,])+mu_N^2
  b_N[i,] = b0+0.5*((E_mu2[i,])*(N+lambda0)-
(2*E_mu[i,]*((N*mean(x))+(lambda0*mu0)))+((lambda0*(mu0^2))+sum(x^2)))
  q_tau[i,] = rgamma(1,a_N,b_N[i,])
  E_tau[i,] = a_N/b_N[i,]

}
```

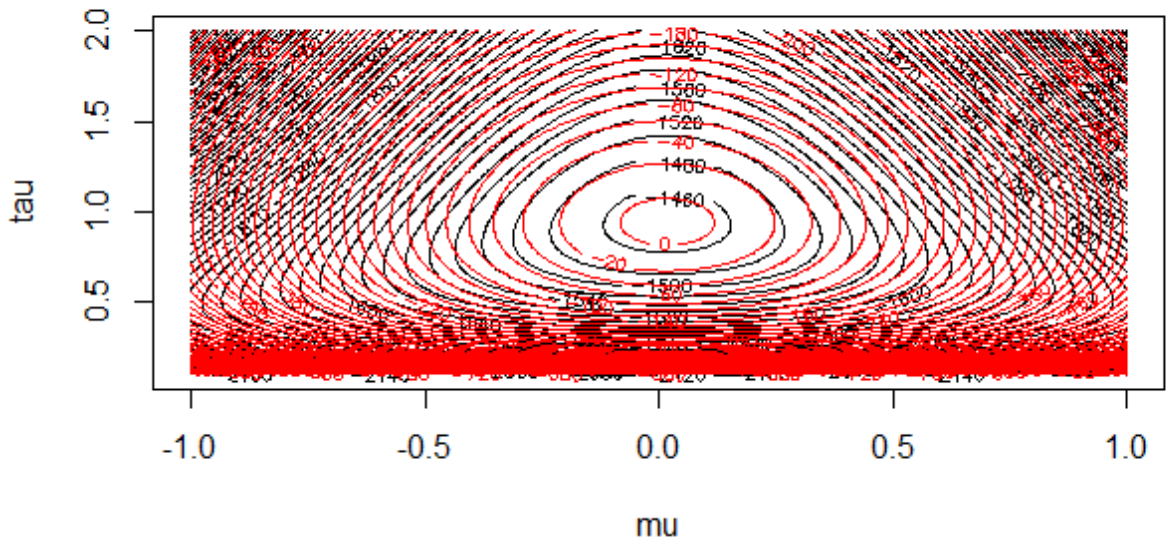

Fig 1. Illustration of variational inference for the mean μ and precision τ of a univariate Gaussian distribution. Contours of the true posterior distribution $p(\mu, \tau|\boldsymbol{y})$ are shown in black. Contours of the optimal factorized approximation, to which the iterative scheme converges, are shown in red.

II) **HIERARCHICAL GAUSSIAN**

This dataset is obtained from GELMAN *et al.* 2004; pp 119-120, pertaining to the study performed for the Educational Testing Services to analyze the effects of coaching programs on SAT scores in each of eight high schools. The outcome variable in each study was a score on a special administration of the SAT-V. The estimated The results are summarized in the following table:

Table 1. Data set for the SAT-V scores from 8 schools

School	Estimated treatment effect, y_j	Standard error of effect estimate, σ_j
A	28	15
B	8	10
C	-3	16
D	7	11
E	-1	9
F	1	11
G	18	10
H	12	18

The following example is taken from GELMAN *et al.* 2004 pp 289, 329, and 332-336. We assume that the standard deviation, σ_j is known. The estimated effects, denoted by y_j, can be interpreted as $\bar{y}_{.j}$ where each y_j is assumed to have normal sampling distributions with relatively large sample sizes (over 30). Here, we label the eight school effects as α_j. The full vector of parameters θ then has 10 dimensions corresponding to $\alpha_1, \alpha_2, \ldots, \alpha_8, \mu, \tau$. Correspondingly, the log posterior density is given by:

$$\log p(\boldsymbol{\theta}|\boldsymbol{y}) = -\frac{1}{2}\sum_{j=1}^{8}\frac{\left(y_j-\alpha_j\right)^2}{\sigma_j^2} - 8log\pi - \frac{1}{2\tau^2}\sum_{j=1}^{8}\left(\alpha_j-\mu\right)^2 + const. \qquad [9]$$

A) <u>Gibbs sampler</u>

The joint posterior density is the same as [9] and we assume that the posterior variances σ_j^2 are known and are given in Table 1. The next step is to find the conditional posterior distribution of each of the parameters.

<u>For α_j:</u>

- The conditional posterior distribution of each α_j given the other parameters in the model is:

$$\alpha_j|\mu,\tau,\boldsymbol{y} \sim N\left(\widehat{\alpha_j}, V_{\alpha_j}\right)$$

$$= N\left(\frac{\left(\frac{y_j}{\sigma_j^2}+\frac{\mu}{\tau^2}\right)}{\frac{1}{\sigma_j^2}+\frac{1}{\tau^2}}, \frac{1}{\frac{1}{\sigma_j^2}+\frac{1}{\tau^2}}\right) \qquad [10]$$

<u>For μ:</u>

- The conditional posterior distribution of μ given $\boldsymbol{y}$ and the other parameters in the model is:

$$\mu|\boldsymbol{\alpha}, \tau, \boldsymbol{y} \sim N\left(\hat{\mu}, \frac{\tau^2}{8}\right)$$

$$= N\left(\frac{\sum_{j=1}^{8} \alpha_j}{8}, \frac{\tau^2}{8}\right)$$

[11]

For τ^2:

- The conditional posterior distribution of τ^2 given the other parameters in the model is a scaled-inverse chi-squared distribution:

$$\tau^2|\boldsymbol{\alpha}, \mu, \boldsymbol{y} \sim Inv - \chi^2(J-1, \hat{\tau}^2)$$

$$= Inv - \chi^2\left(7, \sum_{j=1}^{8} \frac{(\alpha_j - \mu)^2}{7}\right)$$

[12]

```
## Gibbs sampler
# SAT-V scores
y = c(28,8,-3,7,-1,1,18,12)
sigma = c(15,10,16,11,9,11,10,18)
######################

J=8
alpha.update <- function (){
 alpha.hat <- (mu/tau^2 + y/sigma^2)/(1/tau^2 + 1/sigma^2)
 V.alpha <- 1/(1/tau^2 + 1/sigma^2)
 rnorm (J, alpha.hat, sqrt(V.alpha))
}
mu.update <- function (){
 rnorm (1, mean(theta), tau/sqrt(J))
}
tau.update <- function (){
 sqrt(sum((theta-mu)^2)/rchisq(1,J-1))
}
```

We generate independent Gibbs sampling sequences of length 1000 and initialize them based on the range of the data $\boldsymbol{y}$**.** We then run the Gibbs sampler and eventually save the posterior simulation draws for α, μ, τ.

```
n.chains <- 5
n.iter <- 1000
```

```
sims <- array (NA, c(n.iter, n.chains, J+2))
dimnames (sims) <- list (NULL, NULL,
                  c (paste ("alpha[", 1:8, "]", sep=""), "mu", "tau"))
for (m in 1:n.chains){
 mu <- rnorm (1, mean(y), sd(y))
 tau <- runif (1, 0, sd(y))
 for (t in 1:n.iter){
  alpha <- alpha.update ()
  mu <- mu.update ()
  tau <- tau.update ()
  sims[t,m,] <- c (alpha, mu, tau)
 }
}
```

B) <u>EM algorithm</u>

The EM algorithm is an iterative method for finding the mode of the marginal posterior density and is useful for many common models for which it is difficult to maximize the marginal density directly but is easier to work with conditional posterior density. Because of the conjugacy of the normal model, it is easy to perform conditional maximization on the joint posterior density, updating each parameter by its conditional mode. We have already determined the conditional posterior density functions in computing the Gibbs sampler so the conditional modes are easy to compute. We simply need a starting guess for the parameters. Maximizing the conditional posterior densities in equations [10-12], we get our conditional modes. For this small example, the algorithm only needs three iterations to converge to the optimum solution.

To obtain the mode of the posterior density, we average over the parameter $\boldsymbol{\alpha}$ in the E-step and maximize over τ, μ in the M-step. The logarithm of the joint posterior density of all parameters is the same as expression [9].

E step: Averaging over α requires determining the conditional posterior expectations $E_{old}(\alpha_j - \mu)^2$ and $E_{old}(y_j - \mu)^2$, which can be obtained using the conditional posterior distribution from equations [18-19].

$$\begin{aligned} E_{old}(\alpha_j - \mu)^2 &= E((\alpha_j - \mu)^2 \mid \mu^{old}, \tau^{old}, \boldsymbol{y}) \\ &= E_{old}(\alpha_j - \mu)^2 + var_{old}(\alpha_j) \\ &= (\hat{\alpha_j} - \mu)^2 + V_{\alpha_j} \end{aligned}$$

Likewise,

$$E_{old}(y_j - \alpha_j)^2 = (y_j - \hat{\alpha_j})^2 + V_{\alpha_j}$$

Here, $\hat{\alpha_j}$ and V_{α_j} are computed based on $(\mu, \log\tau)^{old}$

M step: We now maximize $E_{old} \log p(\boldsymbol{\alpha}, \mu, \tau|\boldsymbol{y})$ as a function of $\mu, log\tau$. The maximizing values are $\mu^{new}, log\tau^{new}$ with $(\mu, \tau)^{new}$ obtained by maximizing equation [9]:

$$\mu^{new} = \frac{1}{8}\sum_{j=1}^{8} \hat{\alpha}_j$$

$$\tau^{new} = \left(\frac{1}{7} \sum_{j=1}^{8} \left(\left(\hat{\alpha}_j - \mu^{new} \right)^2 + V_{\alpha_j} \right) \right)^{\frac{1}{2}}$$

```
#EM
mu_old = rnorm(1)
tau_old = runif(1)
reps = 100
mu = tau = matrix(,reps,1)
mu[1,] = mu_old
tau[1,] = tau_old
alpha_hat = V_alpha_hat = matrix(,reps,8)

for(i in 2:reps){
 alpha_hat[i-1,] = ((y/sigma^2)+(mu[i-1,]/tau[i-1,]^2))/((1/sigma^2)+(1/tau[i-1,]^2))
 V_alpha_hat[i-1,] = 1/((1/sigma^2)+(1/tau[i-1,]^2))
 mu[i,] = (1/8)* sum(alpha_hat[i-1,])
 tau[i,] = sqrt((1/7)*sum(((alpha_hat[i-1,]-mu[i,])^2)+V_alpha_hat[i-1,]))
}
i=101
alpha_hat[i-1,] = ((y/sigma^2)+(mu[i-1,]/tau[i-1,]^2))/((1/sigma^2)+(1/tau[i-1,]^2))
V_alpha_hat[i-1,] = 1/((1/sigma^2)+(1/tau[i-1,]^2))

alpha_EM = alpha_hat
mu_EM = mu
tau_EM = tau
```

C) VB Algorithm:

Following standard VB practice, we approximate $p(\theta)$by a product of independent densities, denoted by $g(\theta)$, where

$$g(\theta) = g(\alpha_1, \alpha_2, \dots, \alpha_8, \mu, \tau) = g(\alpha_1) g(\alpha_2) \dots g(\alpha_8) g(\mu) g(\tau) \qquad [13]$$

1) Determine the form of approximating distributions and conditional expectations

For α_j:

- For each α_j, we look at $E\ log\ p$, averaging over the seven $\alpha's, \mu, and\ \tau$; i.e., we average over all parameters other than α_j in [9].
- Expectations that do not involve α_j are swept into the constant term

$$E_{g_{-\alpha_j}} \log p(\boldsymbol{\theta}|\boldsymbol{y}) = -\frac{1}{2}\frac{(y_i-\alpha_j)^2}{\sigma_j^2} - \frac{1}{2} E\left(\frac{1}{\tau^2}\right) E\left(\left(\alpha_j - \mu\right)^2\right) + const.$$

[14]

This can be identified as a quadratic form in α_j and $\exp(E\ log\, p(\boldsymbol{\theta}|\boldsymbol{y}))$ is proportional to a normal density when considered as a function of α_j. Upon completing the square and solving,

$$g(\alpha_j) = N\left(\alpha_j \middle| \frac{\left(\frac{y_j}{\sigma_j^2} + E\left(\frac{1}{\tau^2}\right)E(\mu)\right)}{\frac{1}{\sigma_j^2} + E\left(\frac{1}{\tau^2}\right)}, \frac{1}{\frac{1}{\sigma_j^2} + E\left(\frac{1}{\tau^2}\right)}\right)$$

$$g(\alpha_j) = N\left(\alpha_j \middle| M(\alpha_j), S^2(\alpha_j)\right) \quad [15]$$

For μ:

- We average over all parameters other than μ in [9].

$$E_{g_{-\mu}} \log p(\boldsymbol{\theta}|\boldsymbol{y}) = -\frac{1}{2}E\left(\frac{1}{\tau^2}\right)E\left(\left(\alpha_j - \mu\right)^2\right) + const. \quad [16]$$

Again, this can be identified as the logarithm of a normal density. Upon solving,

$$g(\mu) = N\left(\mu \middle| \frac{1}{8}\sum_{j=1}^{8} E(\alpha_j), \frac{1}{8E\left(\frac{1}{\tau^2}\right)}\right)$$

$$g(\mu) = N(\mu|M(\mu), S^2(\mu)) \quad [17]$$

For τ:

- We average over all parameters other than τ in [9].

$$E_{g_{-\tau}} \log p(\boldsymbol{\theta}|\boldsymbol{y}) = -8Elog\pi - \frac{1}{2}E\left(\frac{1}{\tau^2}\right)\sum_{j=1}^{N}\left(\alpha_j - \mu\right)^2 + const. \quad [18]$$

This can be identified as an inverse-gamma function with parameters given as follows:

$$g(\tau^2) = Inv - \chi^2\left(\tau^2 \middle| 7, \frac{1}{7}\sum_{j=1}^{8} E\left(\left(\alpha_j - \mu\right)^2\right)\right)$$

$$g(\tau^2) = Inv - \chi^2\left(\tau^2 \middle| 7, M^2(\tau)\right)$$

$$g(\tau^2) = Inv - \chi^2\left(\tau^2 \middle| 7, \frac{1}{7}\sum_{j=1}^{8}\left(\left(M_{\alpha_j} - M_\mu\right)^2 + S_{\alpha_j}^2 + S_\mu^2\right)\right) \quad [19]$$

These expressions are identical to the derivations of the conditional distributions for the Gibbs sampler and EM algorithm.

2) Starting values
   - Initialize not the parameters α, μ, τ but the parameters in the distributions $g(\alpha_j), g(\mu), g(\tau)$.
   - Draw unbounded parameters $M(\alpha_j), M(\mu)$ from independent $N(0,1)$ distributions
   - Draw bounded parameters S_{α_j}, S_μ from independent U(0,1) distributions

3) Running the algorithm
   - Compute parameters in [12], [14], [16] by plugging in the expectations initialized in the previous step
   - Label the newly computed means and standard deviations as the updated $M's$ and $S's$.
   - The algorithm thus is very similar to EM, with the difference that it is the distributions, rather than point estimates that are being updated.

4) Check improvement of fit
   - The Kullback-Leibler divergence should decrease in each step of VB
   - We can evaluate this expression analytically in this example
   - We ignore constants that don't depend on our parameters of interest
   - The KL divergence is given by:

$$KL(g||p) = -E_g\left(\frac{\log p(\boldsymbol{\theta}|\boldsymbol{y})}{g(\theta)}\right)$$
$$= -E_g(\log p(\boldsymbol{\theta}|\boldsymbol{y})) + E_g(g(\boldsymbol{\theta}))$$
$$= -\frac{1}{2}\sum_{j=1}^{8}\frac{(\boldsymbol{y}-M_\alpha)^2 + S_\alpha^2}{\sigma_j^2} - 8logM_\tau + \frac{1}{2}\sum_{j=1}^{8}\frac{(M_\alpha - M_\mu)^2 + S_\alpha^2 + S_\mu^2}{M_\tau^2}$$
$$+logS_\alpha - logS_\mu - 10logM_\tau + const \qquad [20]$$

5) Comparing variational and full Bayes solutions
   - This variational fit does not allow for dependence among the $\alpha_j's$
   - However, the approximation fits the marginal distribution well in this case
   - VB represents a fast and scalable approach for inference in this problem with large datasets
   - It makes sense to compare the VB solution to the actual posterior density obtained using MCMC.

```
# VB
library(geoR)
set.seed(100)
M_alpha0 = S2_alpha0 = matrix(,8,1)
for(i in 1:8){
  M_alpha0[i,] = rnorm(1)
  S2_alpha0[i,] = runif(1)
}
M_mu0 = rnorm(1)
S2_mu0 = (runif(1))^2
M2_tau0 = 0
```

```
for(i in 1:8)
  M2_tau0 = M2_tau0+((M_alpha0[i,]-M_mu0)^2+S2_alpha0[i,]+S2_mu0)
M2_tau0 = M2_tau0/7

reps=100
alpha = matrix(,reps+1,8)
mu = tau2 =  KL = matrix(,reps+1,1)

M_alpha = S2_alpha = matrix(,reps+1,8)
M_mu = S2_mu = M2_tau = matrix(,reps+1,1)

  M_alpha[1,] = as.vector(M_alpha0)
  M_mu[1,] = M_mu0
  S2_alpha[1,] = as.vector(S2_alpha0)
  S2_mu[1,] = S2_mu0
  M2_tau[1,] = M2_tau0

for(j in 1:reps){
  for(i in 1:8)
    alpha[j+1,i] = rnorm(1,M_alpha[j,i],sqrt(S2_alpha[j,i]))
  mu[j+1,] = rnorm(1,M_mu[j,],sqrt(S2_mu[j,]))
  tau2[j+1,] = rinvchisq(1,7,M2_tau[j,])

  KL[j,] = sum((((y-M_alpha[j,])^2)+S2_alpha[j,])/(2*sigma^2))+(8*log(sqrt(M2_tau[j,])))+
    sum((((M_alpha[j,]-M_mu[j,])^2)+S2_alpha[j,]+S2_mu[j,])/(2*M2_tau[j,]))
  -sum(log(sqrt(S2_alpha[j,])))-log(sqrt(S2_mu[j,]))-(10*log(sqrt(M2_tau[j,])))

  M_alpha[j+1,] = (((y/sigma^2)+((1/M2_tau[j,])*M_mu[j,]))/((1/sigma^2)+(1/M2_tau[j,])))
  S2_alpha[j+1,] = 1/((1/sigma^2)+(1/M2_tau[j,]))
  M_mu[j+1,] = mean(M_alpha[j+1,])
  S2_mu[j+1,] = (1/8)*(1/(1/M2_tau[j,]))
  tmp = 0
  for(k in 1:8)
    tmp = tmp+((M_alpha[j+1,k]-M_mu[j+1,])^2+S2_alpha[j+1,k]+S2_mu[j+1,])
  M2_tau[j+1,] = tmp/7
  }
```

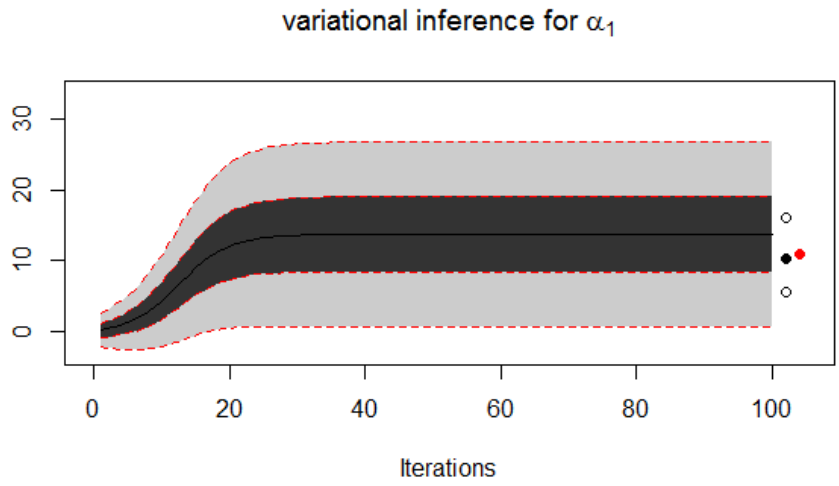

Fig 2. Progress of inferences for the effects in school A for 100 iterations of VB. The lines and shaded regions show the median, 50% interval and 90% interval for the variational distribution. Shown to the right of each graph are the median and the 25 and 75% quantiles for the full Bayes inference computed using the Gibbs sampler (in black) and the estimated parameter value obtained using the EM algorithm (in red).

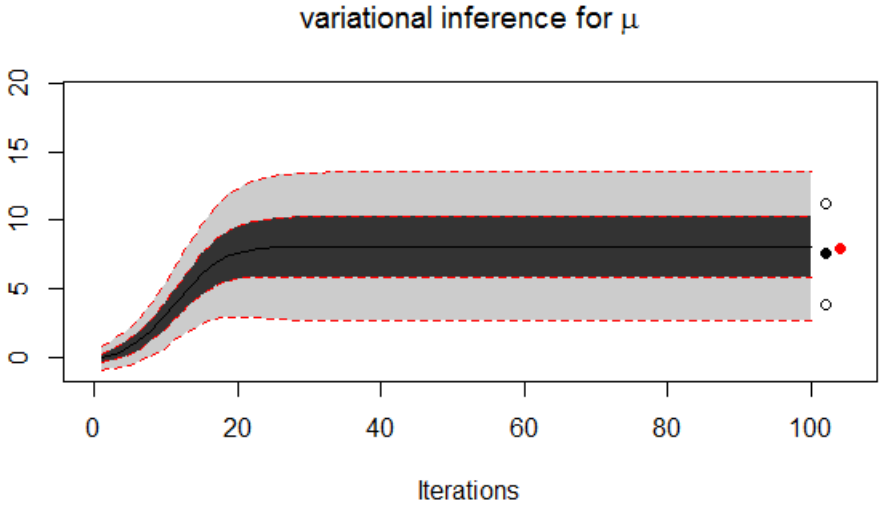

Fig 3. Progress of inferences for the mean parameter μ of all 8 school effects for 100 iterations of VB. The lines and shaded regions show the median, 50% interval and 90% interval for the variational distribution. Shown to the right of each graph are the median and the 25 and 75% quantiles for the full Bayes inference computed using the Gibbs sampler (in black) and the estimated parameter value obtained using the EM algorithm (in red).

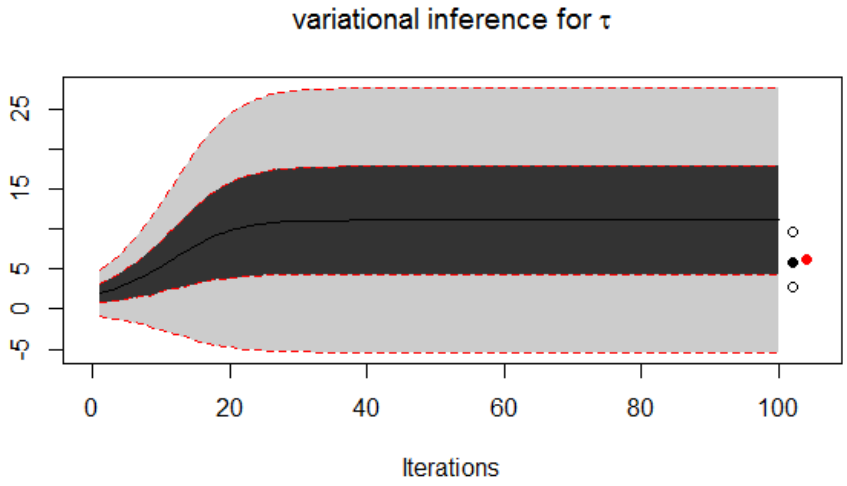

Fig 4. Progress of inferences for the standard deviation parameter τ of all 8 school effects for 100 iterations of VB. The lines and shaded regions show the median, 50% interval and 90% interval for the variational distribution. Shown to the right of each graph are the median and the 25 and 75% quantiles for the full Bayes inference computed using the Gibbs sampler (in black) and the estimated parameter value obtained using the EM algorithm (in red).

ARAKAWA A., 2014 Variational Bayesian Method to Estimate Variance Components. In: *10th World Congress on Genetics Applied to Livestock Production*, Asas.
ATTIAS H., 2000 A Variational Bayesian Framework for Graphical Models
BISHOP C. M., 2006 *Pattern Recognition and Machine Learning*. Springer.
BLEI D. M., JORDAN M. I., 2006 Variational inference for Dirichlet process mixtures. Bayesian Anal. **1**: 121–143.
CARBONETTO P., STEPHENS M., 2012 Scalable Variational Inference for Bayesian Variable Selection in Regression, and Its Accuracy in Genetic Association Studies. Bayesian Anal. **7**: 73–108.
COVER T. M., THOMAS J. A., 2012 *Elements of Information Theory*. John Wiley & Sons.
GELMAN A., CARLIN J. B., RUBIN D. B., STERN H. S., 2004 Bayesian data analysis.
HAYASHI T., IWATA H., 2013 A Bayesian method and its variational approximation for prediction of genomic breeding values in multiple traits. BMC Bioinformatics **14**: 34.
HERMOSILLO G., CHEFD'HOTEL C., FAUGERAS O., 2002 Variational Methods for Multimodal Image Matching. Int. J. Comput. Vis. **50**: 329–343
JAAKKOLA T. S., 2001 Tutorial on Variational Approximation Methods.
JORDAN M. I., GHAHRAMANI Z., JAAKKOLA T. S., SAUL L. K., 1999 An Introduction to Variational Methods for Graphical Models. Mach. Learn. **37**: 183–233.
LI Z., SILLANPÄÄ M. J., 2012 Estimation of quantitative trait locus effects with epistasis by variational Bayes algorithms. Genetics **190**: 231–49.
LOGSDON B. A., HOFFMAN G. E., MEZEY J. G., 2010 A variational Bayes algorithm for fast and accurate multiple locus genome-wide association analysis. BMC Bioinformatics **11**: 58.

LOH P.-R., TUCKER G., BULIK-SULLIVAN B. K., VILHJÁLMSSON B. J., FINUCANE H. K., *et al.*, 2015 Efficient Bayesian mixed-model analysis increases association power in large cohorts. Nat. Genet. **47**: 284–290.
RAJ A., STEPHENS M., PRITCHARD J. K., 2014 fastSTRUCTURE: Variational Inference of Population Structure in Large SNP Data Sets. Genetics **197**: 573–89.